Centro Federal de Educação Tecnológica de Minas Gerais
Programa de Pós-Graduação em Modelagem Matemática e Computacional

Tese de Doutorado

ALGUNS RESULTADOS SOBRE SEPARADORES DE VÉRTICES EM GRAFOS

Sérgio Henrique Nogueira

Orientador: Vinicius Fernandes dos Santos

Belo Horizonte
Maio de 2020

SOME RESULTS ON VERTEX SEPARATORS IN GRAPHS

Sérgio Henrique Nogueira

Advisor: Vinicius Fernandes dos Santos

Belo Horizonte
May, 2020

Sérgio Henrique Nogueira

ALGUNS RESULTADOS SOBRE SEPARADORES DE VÉRTICES EM GRAFOS

Tese apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para obtenção do título de Doutor em Modelagem Matemática e Computacional

Orientador: Vinicius Fernandes dos Santos

Belo Horizonte
2020

Nogueira, Sérgio Henrique
N778a Alguns resultados sobre separadores de vértices em grafos / Sérgio Henrique Nogueira. – 2020.
56 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Vinícius Fernandes dos Santos.
Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Teoria dos grafos – Teses. 2. Análise numérica – Teses.4. Complexidade computacional – Teses. I. Santos, Vinícius Fernandes dos. II. Centro Federal de Educação Tecnológica de Minas Gerais. III. Título.

CDD 511.5

Elaboração da ficha catalográfica pela Biblioteca-Campus II / CEFET-MG

ALGUNS RESULTADOS SOBRE SEPARADORES DE VÉRTICES EM GRAFOS

Sérgio Henrique Nogueira

TESE SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL DO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM MODELAGEM MATEMÁTICA E COMPUTACIONAL.

EXAMINADA POR

Prof. Vinicius Fernandes dos Santos, D.Sc.

Prof. Anísio Mendes Lacerda, D.Sc.

Prof. Flávio Vinicius Cruzeiro Martins, D.Sc.

Guilherme de Castro Mendes Gomes, D.Sc.

Prof. Hugo de Holanda Cunha Nóbrega, D.Sc.

Prof. Phablo Fernando Soares Moura, D.Sc.

Ao pai e à mae.
E à Nathália.

Agradecimentos

Agradeço a Deus, por ter-me concedido o dom da fé, sem a qual acho que eu não teria vencido tantos momentos difíceis nesses quatro anos e ainda finalizado este doutorado.

Agradeço muito à minha família por acreditar sempre em mim, especialmente nas coisas em que eu sempre acreditei. Ao pai e à mãe, que eu tenho certeza, estão muito felizes com este doutorado. A toda minha família, especialmente a Jack, que sempre esteve junto comigo em todos os momentos, especialmente nos mais difíceis, sempre com uma palavra de conforto. Aos meus irmãoss que puderam estar mais próximos de mim nesses anos: Renato (companheiro de sempre e todas as horas), Bira, Sandra (com Bárbara, Nathália e Gabriela).... E aos que não puderam estar próximos, mas apoiaram e torceram sempre. A todos, muito obrigado.

Agradeço à Nathália, minha filha, por todo amor e por sempre me fazer ter forças e seguir em frente.

Agradeço especialmente à Nayle, uma grande companheira em todos os momentos, que soube ter paciência e me apoiar sempre, com muito amor. Muito obrigado por tudo.

Agradeço a sociedade brasileira, que proporcionou toda a minha formação em instituições públicas. Especialmente sou grato à UFV, USP e CEFET MG.

Agradeço ao CEFET pela oportunidade e à UFV por ter concedido a licença para a realização deste doutorado.

Por fim, agradeço de modo especial ao meu orientador, Vinicius Fernandes dos Santos, que com muita competência e dedicação soube compreender minhas limitações e sempre me incentivar no caminho da pesquisa. Você fica registrado para mim como um exemplo profissional. Muito obrigado.

Aos meus amigos, dos vários lugares: da UFV, da roça, do bar, da vida.... A todos, muito obrigado pelo apoio e por tantas vezes ajudarem a tornar a vida mais leve.

Resumo

Separadores de vértices são úteis para resolver uma grande variedade de problemas em grafos. Nesta tese nós estudamos uma caracterização de algumas subclasses de grafos através de separadores de vértices e também estudamos reconfiguração de separadores, sob certas condições.

Na primeira parte, nós estudamos subclasses de grafos cordais definidas por restrições impostas nas relações de continência e interseção de seus separadores minimais de vértices e caracterizamos essas subclasses por subgrafos induzidos proibidos.

Na segunda parte, consideramos as regras mais comuns de reconfiguração e provamos resultados de equivalência e complexidade para Reconfiguração de Separadores de Vértices.

Palavras-chave: Grafos cordais, separadores de vértices, subgrafos induzidos proibidos, reconfiguração, complexidade.

Abstract

Vertex Separators are useful for solving a variety of graph problems. In this thesis we study a characterization of some subclasses of graphs through vertex separators and the reconfiguration of vertex separators, under certain conditions.

In the first part, we study subclasses of chordal graphs defined by restrictions imposed on the containment and intersection relationships of its minimal vertex separators and characterizing them by forbidden induced subgraphs.

In the second part, we study Vertex Separator Reconfiguration problems, under the most common rules of reconfiguration and we provide complexity results for Vertex Separator Reconfiguration.

Keywords: Chordal graphs, vertex separators, forbidden induced subgraphs, reconfiguration, complexity.

Contents

Chapter 1

Introduction

A graph is a ordered pair $(V(G), E(G))$ whose first set $V(G)$ is called vertices set and the second set $E(G)$ is the edges set. Due to this simplicity of representation, graphs are useful to model an infinity of problems of theoretical and practical nature in various research areas.

A separator in a graph is that of a set of vertices that separates the graph into two or more connected components. This idea can be used to guide decomposition thecniques that break the graph into smaller subgraphs in which the solution of a problem, which is complex in the larger graph, can be found more easily in the smaller subgraph.

In the study of graphs, many times it is interesting to consider, given two vertices of the graph, whether it is possible to go from one of these vertices to the other walking on edges; if this is possible we say that there exists a path between these vertices and when there exists a path between every pair of vertices the graph is connected. The concept of separator has its origin in the opposite idea of connectivity: if on the one hand it is important to study the connectivity of a graph, on the other hand it is equally important to study the possibility of keeping two vertices always isolated from one another in a graph, that is, given two vertices of a graph, knowing which sets of vertices separate these two vertices from each other. Given two vertices of a graph, a set of vertices of the graph is a separator of those vertices if the removal of this set makes impossible the existence of a path between them. The use of separators in graphs as a structural tool plays an important role as modern research topic in graph theory, including many algorithms as it can be seen, for example, in [4], [45] and [57].

Rendl and Satirov [66] study vertex separator sets in graphs and cite some different fields in which the problem arises: Very Large Scale Integration fabrication technology for circuit layout problems [3], bioinformatics for protein folding problem[23] and in solving system of equations [51, 52], in most of these examples Separation is a fundamental tool which enables the use of divide and conquer paradigm. Finding minimal vertex separators is also an important issue in communications network [47] and finite element methods [58].

Separator sets are very useful in many branches of research in Graph Theory and also in several practical problems. As an example, we cite the work *The Game of Cops and Robbers on Graphs* [5] in which the authors model the famous game in graphs. The model consists of cops and robbers moving on the vertices of a graph and the goal is to prevent the robber from escaping; in this case the vertices of the graph that the cops should occupy can be seen as a separator between the robber and a way to escape.

In this thesis, we study vertex separators of a graph in two ways:

- Studying the structure of the set of minimal vertex separators of a graph class and its relationship with the family of forbidden induced subgraphs defining this class;
- Considering two vertex separators in a graph and the possibility to transform one into other, obeying some rules and studying the complexity of these problems in some cases.

1.1 Applications

The structure of vertex separators in a graph is useful in various situations, from structural questions of a graph to questions of a purely practical nature. First, we can consider the importance of vertex separators in the characterization of different graph classes. Second, we can think of practical situations such as keeping two computers disconnected from each other on a network; knowing sets of points of distribution that cannot be simultaneously disconnected, under penalty of leaving a region without energy, gas, telephone or water; as well as several other problems that can be modeled in a graph and that have the restriction of not being able to leave two vertices isolated from each other.

1.1.1 Electrical power distribution

According to data released by Operador Nacional do Sistema Elétrico (ONS) [63], the generation and transmission of electric power in Brazil is made by the Sistema Interligado Nacional (SIN) and the isolated systems of the country, whose operations are coordinated and controlled by ONS, under the supervision and regulation of the

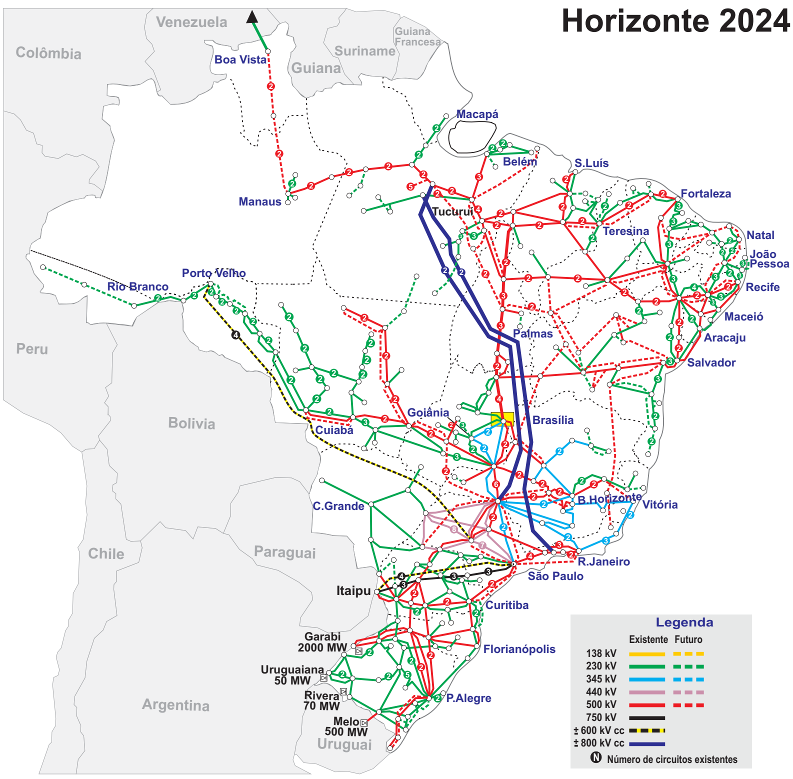


Figure 1.1: Transmission System Map - Horizon 2024 (Source: ONS)

Agência Nacional de Energia Elétrica (Aneel). According to data from 2019, Brazil has 141.756km of basic transmission network, with projection to 181.528km in 2024 (see map of Figure 1.1) representing 99% of the energy consumed in the country. Not wanting to treat the problem in all its complexity, but just to illustrate, we can model the map in a simplified way using a graph, considering the transmission lines as edges and the cities and power plants as vertices. A set of vertices whose simultaneous removal implies a power outage in a locality can be seen as a vertex separator; This set separates the vertex (locality) from all other vertices that may represent an energy supplier. Of course, the problem involves many other variables and many constraints, with much greater complexity than what has been exposed here, but our intention is only to illustrate the use of separators in graphs.

1.1.2 Public Security

To give one more example of using vertex separators in a graph, we now present an analogy of separators parameters applied to a public security context. Imagine that we want to establish a surveilance system that monitors all vehicles that go from Rio de Janeiro to Belo Horizonte by a road and consider all the possibles ways to go from one city to another. If we use Google Maps, for example, considering all secondary roads, we have a very big number of possibilities. To model this in a graph, we can consider each crossing as a vertex and the roads between them must be edges of the graph. We intend to stablish checkpoints on these crossings so that every vehicle travelling from Rio de Janeiro to Belo Horizonte must pass by at least one checkpoint and it is reasonable to think that you want a minimum number of checkpoints. A vertex set of the graph that covers all paths between Rio de Janeiro and Belo Horizonte unchecked is a vertex separator of graph that represents the situation.

1.2 About this thesis

In this thesis we study two questions related to vertex separators: the first one being a structural one andthe latter an algorithmic one.

In the first part we prove a theorem giving a characterization of some subclasses of chordal graphs using forbidden induced subgraphs and minimal separators. This result was presented in EURO/ALIO 2018 International Conference on Applied Combinatorial Optimization and it is in a paper that was published on a special issue of Electronic Notes in Discrete Mathematics (ENDM) associated with the conference in August, 2018 [62].

In the second part we study vertex separators reconfiguration problems and we prove some results about complexity for the usual reconfiguration rules, giving results of NP-hardness and some cases in which the problem can be solved in polynomial time. These results are in a submitted paper [31]. I would like to thank Guilherme C. M. Gomes for his lecture of this chapter.

Chapter 2

Preliminaries

2.1 Definitions and basic considerations

In this chapter we present some important definitions, notation and basic results that we will use throughout the text.

Let $G = (V(G), E(G))$ be a graph with set of vertices $V(G)$ and set of edges $E(G)$. When there is no ambiguity, we denote the graph by $G = (V, E)$ or simply G.

If there exists an edge e in G with $e = uv$ we say that u and v are *adjacent*, u and v are incident to e and e is incident to u, v or e joins u and v.

In the scope of this work, the set V is finite and the elements of $E \subset \binom{V}{2}$. With this we have a finite, undirected and simple graph.

The *degree* of a vertex $v \in G$, denoted by $d(v)$, is the number of edges incident to v in G.

If $V' \subseteq V$ the subgraph $G' = (V', E')$, where $E' = \{uv \in E; u, v \in V'\}$, is called the subgraph of G induced by V', denoted by $G[V']$, and we say that $G[V']$ is an *induced subgraph* of G.

An *isomorphism* of graphs G and H is a bijection between the vertex sets of G and H

$$f : V(G) \to V(H)$$

such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. If there exists a isomorphism between G an H we say that G and H are *isomorphic*.

Given a graph H, we say that G is H-free (or that H is forbidden in G) if no induced subgraph of G is isomorphic to H.

A *path* P of length k in a graph G between two vertices v_0 and v_k is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$, where $v_i v_{i+1} \in E(G)$, for $i = 0, \ldots, k-1$. A *cycle* in a graph G is a path in G whose endpoints v_0 and v_k are the same and a graph is acyclic if it has no induced cycle.

The *distance* between two vertices u and v in a graph G is the length of a shortest path between u and v. The *diameter* of a graph G, denoted by $diam(G)$, is the value of the longest shortest path of G.

A graph G is *connected* if for every pair $u, v \in V(G)$ there exists a path from u to v in G. A *tree* is a connected and acyclic graph.

Given two non-adjacent vertices u and v in the same connected component of G, a *uv-separator* is a set $S \subset V(G)$ such that u and v are in different connected components of $G[V(G) \setminus S]$. This separator S is minimal if no proper subset of S is also a *uv-separator*. We will just say *minimal vertex separator* to refer to a set S that is a minimal uv-separator for some pair of non-adjacent vertices u and v in G.

Given two sets of vertices V_1, V_2 of a graph G we say that a set of vertices S separates V_1 and V_2 if for every pair of vertices u, v with $u \in V_1$ and $v \in V_2$ every path from u to v in G contains a vertex of S.

In a graph G, the *open neighbourhood* of a vertex v is the set $N(v)$ of vertices u of G such that $uv \in E(G)$, that is, $N(v) = \{u \in V(G); uv \in E(G)\}$ and a vertex v is called *simplicial* if the subgraph of G induced by the vertex set $\{v\} \cup N(v)$ is a complete graph.

Given a graph G, the *chromatic number* of G, denoted by $\chi(G)$, is the minimum number of colors needed to assign a color to each vertex of G in such a way that adjacent vertices receive different colours; the *clique number* of G, denoted by $\omega(G)$ is the size of a largest clique of G.

2.2 Graph Classes

A *graph class* is a set of graphs, usually defined by properties that its members satisfy. Some graph classes have been extensively studied over the years by having interesting applications or structures that enable solving problems that are often difficult in general graphs. We will now define two classes that will be essential in Chapters 3 and 4, respectively: chordal and bipartite graphs.

A graph is *chordal* if every cycle of length greater than three has a chord; a chord is an edge connecting two non-consecutive vertices in the cycle. In other words, a graph is chordal if it has no induced cycle of length greater than three.

A *bipartite* graph G is a graph in which the set of vertices can be partitioned into two sets A and B so that each edge has an endpoint in A and the other in B.

2.3 Characterization by forbidden induced subgraphs

When we talk about characterization of a class of graphs we are trying to give a property that all graphs of that class satisfy and, even more, if a graph satisfies that property then it belongs to that particular class.

Characterizing a class of graphs by a family of forbidden induced graphs means giving a set of graphs that do not appear as induced subgraphs in any of the graphs of the class; moreover, if a graph is such that it does not have any of the subgraphs of that set as an induced subgraph, then that graph is in such class.

The characterization of a class of graphs by forbidden subgraphs is an important and very useful method, since it allows in many cases the recognition of the members of the class without having to resort directly to the definition of the class (which often makes it excessively difficult) and in many other cases allows much simpler algorithms for solving certain problems.

As an example, we can think of the class of perfect graphs, introduced by Berge [2]: a graph G is perfect if for every induced subgraph H of G the chromatic number is equal to the clique number, that is, $\chi(H) = \omega(H)$. Chudnovsky et al. [15] proved that a graph is perfect if and only if it does not contain neither an odd cycle of length at least five nor its complement as induced subgraphs,.

Many other characterizations by forbidden subgraphs exist, such as, for example graph split, which can be characterized as *split graph* $= \{2K_2, C_4, C_5\}$*-free* [22] and others as path graphs whose set of forbidden subgraphs is infinite [50]. Some other important examples: a graph is chordal if and only if it is C_i*-free*, for C_i a cycle, $i \geq 4$; and a graph is bipartite if and only if it is $\{C_{2i+1}\}$*-free* for $i \geq 1$, i.e. odd cycle free.

2.4 Reconfiguration

A *Reconfiguration Problem* in graphs consists of, given two sets of vertices of this graphs, that are solutions of a problem or that have same determined property, we can to transform a set into the other is a step-by-step sequence so that each intermediate set has the same initial properties and each step abides by a fixed reconfiguration rule.

It is easy to imagine problems modeled in graphs in witch it is interesting, given two solutions of the problem, one asks if it it is possible to transform a solution into the other in a step-by-step sequence in a way that each intermediate step continues a solution of the problem.

As an example, we consider the problem of Section 1.1.2: suppose that after establishing these checkpoints, for some reason it is established that some of the checkpoints are not satisfactory. In this case, we want to transform this set of checkpoints into another one, but so that throughout the transformation we always have a set of checkpoints that actually separate Belo Horizonte and Rio de Janeiro.

A *Reconfiguration Problem* in graphs consists of, given two solutions of a problem instance, we wish to find a step-by-step transformation between two solutions so that all intermediate results are also solutions of the problem and each step abides by a fixed reconfiguration rule. Reconfiguration problems have been studied for many structures and in [61] we have an important survey about reconfiguration in graph.

Chapter 3

Vertex separators in chordal graphs and forbidden induced subgraphs

3.1 Introduction

In this chapter we present our main results about characterization by forbidden induced subgraphs. We study the relationship between the structure of a graph and its family of minimal vertex separators. In particular we characterize the graph classes arising from properties imposed on the family of minimal vertex separators.

These results were presented in *Characterization by forbidden induced subgraphs of some subclasses of chordal graphs* published in August 2018 [62].

1. First we study the possibilities of combination of the minimal vertex separators of a chordal graph and we prove results that give characterizations of some subclasses of chordal graphs through of these combinations and forbidden induced subgraphs;

2. Second we study the Helly property for the multiset of minimal vertex separators of a chordal graph and we prove a result of characterization when the separators satisfy the Helly property, again using forbidden induced subgraphs.

Our goal is to characterize hereditary subclasses of chordal graphs by the intersection and containment relations of their minimal vertex separators and by forbidden induced subgraphs.

Although a graph in general may have an exponential number of minimal vertex separators, there exist linear-time algorithms to list all the minimal vertex separators of a chordal graph as, for example, [45].

A *clique* is a set of pairwise adjacent vertices. In this chapter and in the next one, we use clique to refer to a maximal clique, that is, clique here is a maximal set of pairwise adjacent vertices. A *clique tree* of a connected graph G is any tree T whose vertices are the cliques of G such that for every two cliques C_i, C_j each clique on the path from C_i to C_j in T contains $C_i \cap C_j$. As we will see in Theorem 3.3 (iv), having a clique tree is an intrinsic property of chordal graphs. It is not difficult to see that in a clique tree T of a chordal graph G, for every vertex v of G the set of vertices of T that correspond to the cliques containing v induces a subtree of T.

Two cliques C_1, C_2 of G form a *separating pair* if $C_1 \cap C_2$ is non-empty, and every path in G from a vertex of $C_1 \setminus C_2$ to a vertex of $C_2 \setminus C_1$ contains a vertex of $C_1 \cap C_2$. Every minimal vertex separator of a chordal graph G is a clique, see Theorem 3.3 (iii) and moreover, it is precisely the intersection of two cliques, as follows.

Theorem 3.1 ([33]). *A set S is a minimal vertex separator of a chordal graph G if and only if there exist cliques C_1, C_2 forming a separating pair such that $S = C_1 \cap C_2$.*

Other graph classes can be characterized by separators, as the *unichord-free* which are characterized as the graphs whose minimal separators are edgeless subgraphs [56]. A graph is *unichord-free* if and only if every minimal separator induces an edgeless subgraph.

Let G be a chordal graph and let $\mathcal{T}(G)$ be a clique tree of G. The edges of $\mathcal{T}(G)$ can be labeled as the intersection of the endpoints, and these labels are exactly the minimal vertex separators. We denote by $\mathbf{S}_{\mathcal{T}}(G)$ the multiset of labels of the edges of $\mathcal{T}(G)$. A graph can have several distinct clique trees but when we are studying chordal graphs we have the next result giving us that the multiset $\mathbf{S}_{\mathcal{T}}(G)$ is independent of the clique tree.

Theorem 3.2 ([7]). *Let G be a chordal graph. The multiset $\mathbf{S}_{\mathcal{T}}(G)$ of minimal vertex separators of G is the same for every clique tree $\mathcal{T}(G)$.*

For the graph of Figure 3.1, we have

$$\mathbf{S}_{\mathcal{T}}(G) = \{\{c\}, \{h\}, \{h\}, \{a, b\}, \{b, c\}, \{b, d\}, \{b, f\}, \{f, h\}\}.$$

In the light of Theorem 3.1 from now on we omit the subscript and use simply $\mathbf{S}(G)$.

Let G be a graph with n vertices. We say that G has a *perfect elimination ordering* if there is an ordering $\{v_1, \ldots, v_n\}$ of vertices of G such that each v_i is simplicial in the subgraph induced by the vertices $\{v_1, \ldots, v_i\}$.

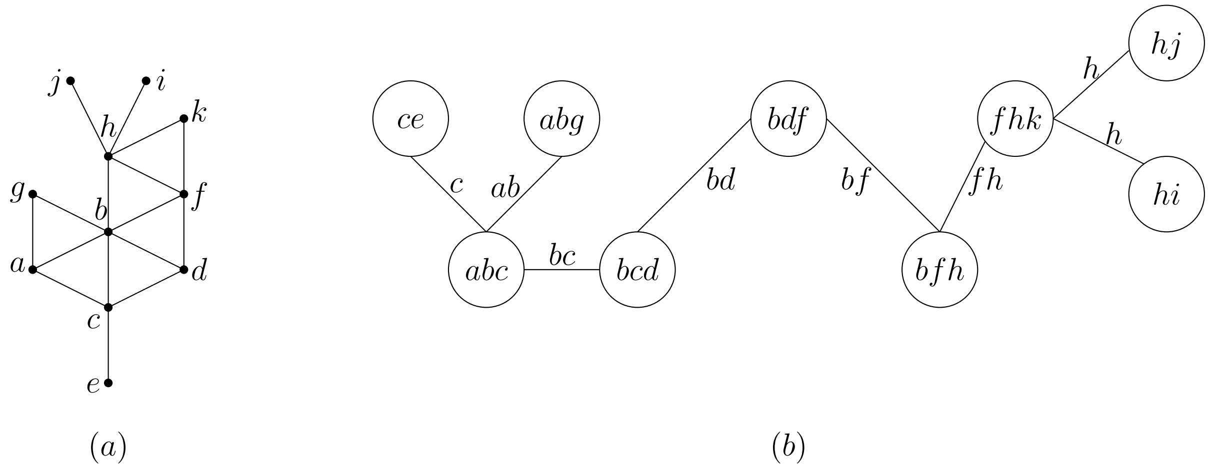


Figure 3.1: (a) A chordal graph G. (b) A clique tree of G.

Chordal graphs have many characterizations, and in this work we are interested especially in the last two points of the following theorem.

Theorem 3.3. *Let G be a connected graph. The following statements are equivalent and characterize a chordal graph:*

- *(i) G has a perfect elimination order ([24]);*
- *(ii) G is the intersection graph of subtrees of a tree ([11], [26] and [75]);*
- *(iii) Every minimal vertex separator of G is a clique ([19]);*
- *(iv) G has a clique tree ([26]).*

As we can see in Theorem 3.3 (iii), minimal vertexseparators and cliques have a strict relation to chordal graphs. In addition, however, the idea of cliques as separator sets appears in several other situations, as we can mention, for example, an important work of Tarjan [72] that gives a decomposition of a graph by clique separators and it shows that this decomposition is useful for many classical problems such as vertex coloring, maximum independent set, among others, in many graph classes. This algorithm was extended for clique minimum separators [48]. In some cases it is useful decomposition in maximal clique separators, as used in [59] to characterize and recognize several related classes of intersection graphs of paths in tree. We note that the algorithm proposed in [72] was incorrect and a correct algorithm was finally proposed by M. Cerioli, H. Nobrega and P. Viana in [14].

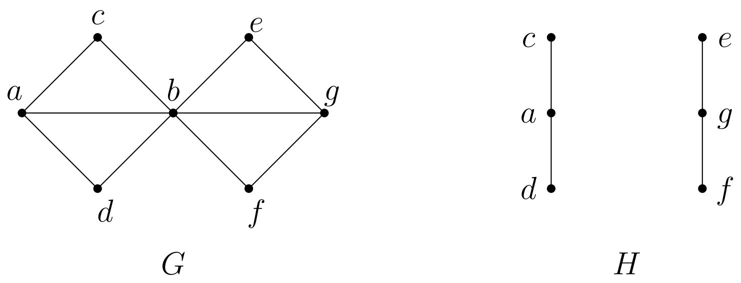


Figure 3.2: $\mathbf{S}(G) = \{\{b\}, \{a, b\}, \{b, g\}\}$ $\mathbf{S}(H) = \{\{a\}, \{g\}\}\}$

To avoid ambiguity, we clarify that for sets R and S, we denote $R \subseteq S$ if R is a subset of S, and $R \subset S$ if R is a proper subset of S.

A graph class $\mathcal{G}$ is hereditary if for every $G \in \mathcal{G}$ and every induced subgraph H of G, $H \in \mathcal{G}$. Not every restriction on the minimal vertex separators leads to a hereditary graph class, as we can see in Figure 3.2. On graph G we have that the set of minimal vertex separators is $\mathbf{S}(G) = \{S_1, S_2, S_3\}$ where $S_1 = \{a, b\}$, $S_2 = \{b\}$, $S_3 = \{b, g\}$ and we have $S_1 \cap S_2 = S_1 \cap S_3 = S_2 \cap S_3 = \{b\} \neq \emptyset$. On the induced subgraph H of G we have the multiset of minimal vertex separators $\mathbf{S}(H) = \{S_4, S_5\}$ where $S_4 = \{a\}$, $S_5 = \{g\}$, and we have $S_3 \cap S_4 = \emptyset$.

In order to obtain a characterization by forbidden induced subgraphs we will impose the additional requirement of being hereditary.

3.2 Characterization

We start with an auxiliary result showing that minimal vertex separators are, in some sense, hereditary. After this, we prove some lemmas with the possibilities of intersection and containment between two minimal vertex separators. With these results we prove a theorem giving the characterization.

Lemma 3.4. *Let G be a chordal graph, let S be a minimal vertex separator of G, $R \subset S$ be a proper subset of S and $S' = S \setminus R$. Then there exist cliques C'_1, C'_2 in $G \setminus R$ such that S' separates C'_1 and C'_2.*

Proof. Since S is a minimal vertex separator of G, there exist cliques C_1, C_2 in G such that $S = C_1 \cap C_2$. Let C'_1, C'_2 be cliques in $G' = G \setminus R$ such that $(C_i \setminus R) \subset C'_i$, $i = 1, 2$. S' separates C'_1 and C'_2, because if there exists a path in G' between a vertex $u_1 \in C'_1$ and $u_2 \in C'_2$ in G', then there exists a path from a vertex of C_1 to a vertex of C_2 in $G \setminus S$, contradicting the fact that S separates C_1 and C_2 in G. Now suppose that

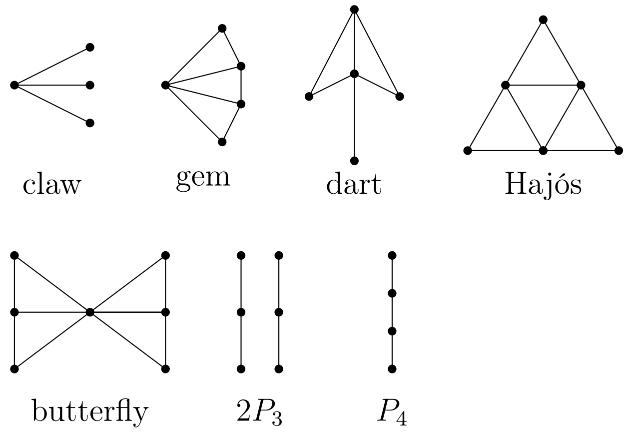


Figure 3.3: Forbidden induced subgraphs considered in this chapter

S' is not minimal and let R' be a non-empty subset of S' such that $S' \setminus R'$ separates C_1', C_2'. Then $S \setminus R'$ separates C_1, C_2, contradicting the fact that S is a minimal vertex separator of G. □

Lemma 3.5. *Let G be a chordal graph. There exist an induced subgraph G' of G and a pair $S_1, S_2 \in \mathbf{S}(G')$ such that $S_1 \cap S_2 = \emptyset$ if and only if G has a P_4 or a $2P_3$ as an induced subgraph.*

Proof. Let G' be an induced subgraph of G. Let $\mathcal{T}'$ be a clique tree of G' and $\mathbf{S}(G')$ be the multiset of minimal vertex separators of G'. Suppose that there exist $S_1, S_2 \in \mathbf{S}(G')$, with $S_1 \cap S_2 = \emptyset$. First, suppose that there exist adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with labels S_1, S_2 and let C_1, C_2, C_3 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_2 \cap C_3$ (See Figure 3.4). Suppose $x \in S_1$ and $y \in S_2$. Since the cliques are maximal there must exist $a \in C_1 \setminus C_2$ and $b \in C_3 \setminus C_2$. Then $\{a, x, y, b\}$ induces a P_4. Now suppose that there exist non-adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with labels S_1, S_2 and let C_1, C_2, C_3, C_4 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_3 \cap C_4$ (see Figure 3.5). Without loss of generality we can consider that the path in $\mathcal{T}'$ from $\{C_1, C_2\}$ to $\{C_3, C_4\}$ contains C_2 and C_3. Suppose $x \in S_1$ and $y \in S_2$. Since the cliques are maximal we must have $a \in C_1 \setminus C_2$, $b \in C_4 \setminus C_3$ and $c \in C_2 \setminus C_1$. If $xy \in E(G')$ then $\{a, x, y, b\}$ is an induced P_4 of G; otherwise if $cy \in E(G')$ then $\{a, x, c, y\}$ induces a P_4; else there exists $d \in C_3 \setminus \{a, b, c, x, y\}$. If $dx \in E(G')$ then $\{x, d, y, b\}$ is P_4; if $cd \in E(G')$ then $\{x, c, d, y\}$ is P_4 else $\{a, x, c\}$, $\{d, y, b\}$ induce $2P_3$.

Conversely, suppose that there exists an induced subgraph G' of G isomorphic to P_4 with vertices v_1, v_2, v_3, v_4 and cliques $C_1 = \{v_1, v_2\}$, $C_2 = \{v_2, v_3\}$ and $C_3 = \{v_3, v_4\}$, see Figure 3.6. Then we have $S_1 = C_1 \cap C_2 = \{v_2\}$, $S_2 = C_2 \cap C_3 = \{v_3\}$ and

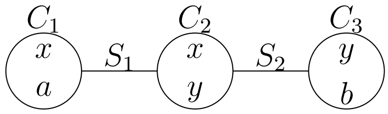


Figure 3.4: Adjacent edges

Figure 3.5: Non-adjacent edges

$S_1 \cap S_2 = \emptyset$. Now suppose we have an induced subgraph G' of G isomorphic to $2P_3$ with vertices $v_1v_2v_3$ and $v_4v_5v_6$ and cliques $C_1 = \{v_1, v_2\}$, $C_2 = \{v_2, v_3\}$, $C_3 = \{v_4, v_5\}$ and $C_4 = \{v_5, v_6\}$. Then we have $S_1 = C_1 \cap C_2 = \{v_2\}$, $S_2 = C_3 \cap C_4 = \{v_5\}$ and $S_1 \cap S_2 = \emptyset$. □

Lemma 3.6. *Let G be a chordal graph. There exist an induced subgraph G' of G and a pair $S_1, S_2 \in \mathbf{S}(G')$ such that $S_1 = S_2$ if and only if G has a claw as an induced subgraph.*

Proof. Let G' be an induced subgraph of G. Let $\mathcal{T}'$ be a clique tree of G', $\mathbf{S}(G')$ be the multiset of minimal vertex separators of G' and suppose that there exist $S_1, S_2 \in \mathbf{S}(G')$, with $S_1 = S_2$. First, suppose that there exist adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with labels S_1, S_2 and let C_1, C_2, C_3 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_2 \cap C_3$. Since the separators are equal, take $x \in S_1 \cap S_2$. Since the cliques are maximal there must exist $a \in C_1 \setminus C_2$, $b \in C_3 \setminus C_2$ and $c \in C_2 \setminus (C_1 \cup C_3)$. Then $\{x, a, b, c\}$ induces a claw. Now suppose that there exist non-adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with labels S_1, S_2 and let C_1, C_2, C_3, C_4 be distinct cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_3 \cap C_4$. Without loss of generality we can consider that every path in $\mathcal{T}'$ from $\{C_1, C_2\}$ to $\{C_3, C_4\}$ contains C_2 and C_3. Let $x \in \bigcap_{i=1,\ldots,4} C_i$. Since the cliques are maximal we must have $a \in C_1 \setminus C_2$, $b \in C_4 \setminus C_3$ and $c \in C_2 \setminus C_1$ and then $\{x, a, b, c\}$ induces a claw.

Conversely let G' be an induced subgraph of G and let $\{x, a, b, c\}$ be an induced claw of G' centered at x, with cliques $C_1 = \{x, a\}, C_2 = \{x, b\}, C_3 = \{x, c\}$, see

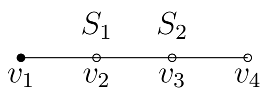


Figure 3.6: P_4

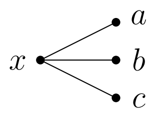


Figure 3.7: Claw

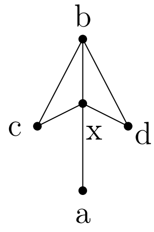


Figure 3.8: Dart

Figure 3.7. Then we have $S_1 = C_1 \cap C_2 = \{x\} = C_2 \cap C_3 = S_2$. □

Lemma 3.7. *Let G be a chordal graph. There exist an induced subgraph G' of G and a pair $S_1, S_2 \in \mathbf{S}(G')$ such that $S_1 \subset S_2$ or $S_2 \subset S_1$ if and only if G has a dart as an induced subgraph.*

Proof. Let G' be an induced subgraph of G. Let $\mathcal{T}'$ be a clique tree of G' and $\mathbf{S}(G')$ be the multiset of minimal vertex separators of G' and suppose that $\exists S_1, S_2 \in \mathbf{S}$ with $S_1 \subset S_2$. In this case we can assume that S_1, S_2 are adjacent in $\mathcal{T}'$. Let C_1, C_2, C_3 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_2 \cap C_3$. Let $x \in S_1 \cap S_2$ and $y \in S_2 \setminus S_1$. Since the cliques are maximal there must exist $a \in C_1 \setminus C_2$, $b \in C_2 \setminus C_3$ and $c \in C_3 \setminus C_2$. Note that $b \notin C_1$, because $S_1 \subset S_2$. Therefore $\{a, x, y, b, c\}$ induces a dart.

Conversely let G' be an induced subgraph of G and let $\{a, x, b, c, d\}$ be a induced dart of G', see Figure 3.8. Let $C_1 = \{x, a\}, C_2 = \{x, b, c\}, C_3 = \{x, b, d\}$ be its cliques. Then we have $S_1 = C_1 \cap C_2 = \{x\}$ $S_2 = C_2 \cap C_3 = \{x, b\}$ and $S_1 \subset S_2$. □

Lemma 3.8. *Let G be a chordal graph. There exist an induced subgraph G and a pair $S_1, S_2 \in \mathbf{S}(G')$ such that ($S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$) e $S_1 \cap S_2 \neq \emptyset$ if and only if G has a gem or a butterfly as an induced subgraph.*

Proof. Let G' be an induced subraph of G. Let $\mathcal{T}'$ be a clique tree of G' and $\mathbf{S}(G')$ be the multiset of minimal vertex separators of G' and suppose that exist $S_1, S_2 \in \mathbf{S}(G')$, with $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$. First, suppose that there exist adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with labels S_1, S_2, respectively and let C_1, C_2, C_3 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_2 \cap C_3$. Take $x \in S_1 \cap S_2$, $y \in S_1 \setminus S_2$ and $z \in S_2 \setminus S_1$. Since the cliques are

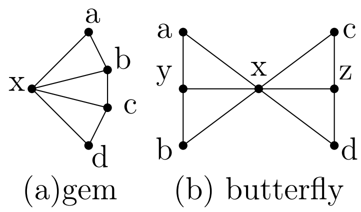


Figure 3.9: (a) Gem and (b) butterfly

maximal there must exist $a \in C_1 \setminus C_2$, $b \in C_3 \setminus C_2$ and $\{x, a, y, z, b\}$ induces a gem. Now suppose that there exist non-adjacent edges $e_1, e_2 \in E(\mathcal{T}')$ with label S_1, S_2 and let C_1, C_2, C_3, C_4 be cliques such that $S_1 = C_1 \cap C_2$ and $S_2 = C_3 \cap C_4$. Let $x, y \in C_1 \cap C_2$ and $x, z \in C_3 \cap C_4$. Then we must have $a \in C_1 \setminus C_2$, $b \in C_4 \setminus C_3$ and $c \in C_2 \setminus C_1$. Note that $xz \in E(G')$. If $cz \in E(G')$ then $\{x, a, y, c, z\}$ induces a gem; else there exists $d \in C_3 \setminus \{a, b, c, x, y, z\}$ such that $\{a, y, c, x, b, z, d\}$ induces butterfly.

Conversely, let G' be an induced subgraph of G and suppose that $\{x, a, b, c, d\}$ is a induced gem of G', with cliques $C_1 = \{x, a, b\}, C_2 = \{x, b, c\}, C_3 = \{x, c, d\}$, see Figure 3.9 (a). Then we have $S_1 = C_1 \cap C_2 = \{x, b\}$, $S_2 = C_2 \cap C_3 = \{x, c\}$ and then $S_1 \cap S_2 \neq \emptyset$ and ($S_1 \nsubseteq S_2$ and $S_2 \nsubseteq S_1$). Now suppose that G' has an induced butterfly $\{x, y, a, b, z, c, d\}$ with cliques $C_1 = \{x, y, a\}, C_2 = \{x, y, b\}, C_3 = \{x, z, c\}, C_4 = \{x, z, d\}$, see Figure 3.9 (b). Then we have $S_1 = C_1 \cap C_2 = \{x, y\}$, $S_2 = C_3 \cap C_4 = \{x, z\}$ and then $S_1 \cap S_2 \neq \emptyset$ and ($S_1 \nsubseteq S_2$ and $S_2 \nsubseteq S_1$). □

Let $\mathcal{H}$ be a hereditary subclass of chordal graphs. Let $G \in \mathcal{H}$, and $\mathbf{S}(G)$ be the multiset of minimal vertex separators of G. For each pair $S_i, S_j \in \mathbf{S}(G)$ one of the following situations holds:

- (a) **Disjunction:** $S_i \cap S_j = \emptyset$.
- (b) **Equality:** $S_i = S_j$.
- (c) **Containment:** $S_i \subset S_j$ or $S_j \subset S_i$.
- (d) **Overlap:** ($S_i \nsubseteq S_j$ and $S_j \nsubseteq S_i$) and $S_i \cap S_j \neq \emptyset$.

Remark 3.9. *Since we are interested in a hereditary class $\mathcal{H}$, we can note that, by Lemma 3.4, if $\mathcal{H}$ is such that given $G \in \mathcal{H}$ we have Containment for every pair $S_i, s_j \in \mathbf{S}(G)$ we must allow Equality and analogously if we have Overlap then we must allow Disjunction.*

Remark 3.10. *And again by heredity we can note that if a class is* claw-free *then it is* dart-free*; if it is* P_4-free *then it is* gem-free *and if it is* dart-free *or* $2P_3$-free *then it is* butterfly-free.

Hence all possible combinations of properties $(a) - (d)$ are characterized in the following theorem.

Theorem 3.11. *Let G be a chordal graph. Then for every G' induced subgraph of G and for every pair $S_i, S_j \in \mathbf{S}(G'), i \neq j$, we have:*

- *(i)* $S_i \cap S_j = \emptyset \Leftrightarrow G$ *is* (claw, gem)-free;
- *(ii)* $S_i = S_j \Leftrightarrow G$ *is* (P_4, gem, butterfly)-free;
- *(iii)* $S_i \cap S_j = \emptyset$ *or* $S_i = S_j \Leftrightarrow G$ *is* (dart, gem)-free;
- *(iv)* $S_i \cap S_j = \emptyset$ *or* $S_i = S_j$ *or* $S_i \subset S_j$ *or* $S_j \subset S_i \Leftrightarrow G$ *is* (gem, butterfly)-free;
- *(v)* $S_i \cap S_j = \emptyset$ *or* $S_i = S_j$ *or* $(S_i \nsubseteq S_j$ *and* $S_j \nsubseteq S_i) \Leftrightarrow G$ *is* dart-free;
- *(vi)* $S_i = S_j$ *or* $S_i \subset S_j$ *or* $S_j \subset S_i \Leftrightarrow G$ *is* $(P_4, 2P_3,)$-free;
- *(vii)* $S_i \cap S_j = \emptyset$ *or* $(S_i \nsubseteq S_j$ *and* $S_j \nsubseteq S_i) \Leftrightarrow G$ *is* claw-free.

Proof.

- (i) $\forall S_i, S_j\ S_i \cap S_j = \emptyset \Leftrightarrow G$ is *(claw, gem)-free.*

($\Rightarrow$) $\forall S_i, S_j\ S_i \cap S_j = \emptyset \Rightarrow G$ is (claw, gem)-free.
Let $\{x\}\{a, b, c\}$ be an induced claw of G', with cliques $C_1 = \{x, a\}, C_2 = \{x, b\}, C_3 = \{x, c\}$. Then we have $S_1 = C_1 \cap C_2 = \{x\} = C_2 \cap C_3 = S_2$, and $S_1 \cap S_2 \neq \emptyset$, contradiction. If G has an induced gem, it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \cap S_2 \neq \emptyset$, contradiction again.

($\Leftarrow$) G is *(claw, gem)-free* $\Rightarrow \forall S_i, S_j\ S_i \cap S_j = \emptyset$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \cap S_2 \neq \emptyset$. We have three cases:

1. $S_1 = S_2$

By Lemma 3.6, G has a claw.

2. $S_1 \subset S_2$ or $S_2 \subset S_1$

By Lemma 3.7, G has a dart.

3. $S_1 \nsubseteq S_2$ and $S_2 \nsubseteq S_1$

By Lemma 3.8, G has a gem or a butterfly.

By Remark 3.10, this is equivalent to say that G has a claw or a gem. Contradiction. Therefore $S_i \cap S_j = \emptyset \Leftrightarrow G$ is *(claw, gem)-free.*

- (ii) $\forall S_i, S_j \ S_i = S_j \Leftrightarrow G$ is *(P_4, $2P_3$, dart)-free.*

($\Rightarrow$) $\forall S_i, S_j \ S_i = S_j \Rightarrow G$ is *(P_4, $2P_3$, dart)-free.*
If G has an induced P_4 or $2P_3$ or dart then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \neq S_2$, contradiction.

($\Leftarrow$) G is *(P_4, $2P_3$,dart)-free* $\Rightarrow \forall S_i, S_j \ S_i = S_j$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \neq S_2$. We have three cases:

1. $S_1 \cap S_2 = \emptyset$

By Lemma 3.5, G has a P_4 or a $2P_3$, contradiction.

2. $S_1 \subset S_2$ or $S_2 \subset S_1$

By Lemma 3.7, G has a dart, contradiction.

3. $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$

By Lemma 3.8, G has a gem or a butterfly; by Remark 3.10 it has a P_4 or it has a $2P_3$ and a dart, contradiction.

Therefore $S_i = S_j \Leftrightarrow G$ is *(P_4, $2P_3$, dart)-free.*

- (iii) $\forall S_i, S_j \ S_i \cap S_j = \emptyset$ or $S_i = S_j \Leftrightarrow G$ is *(dart, gem)-free.*

($\Rightarrow$) $\forall S_i, S_j \ S_i \cap S_j = \emptyset$ or $S_i = S_j \Rightarrow G$ is *(dart, gem)-free.*
If G has a dart, then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \neq S_2$ and $S_1 \cap S_2 \neq \emptyset$, contradiction. Analogously if it has a gem.

($\Leftarrow$) G is *(dart, gem)-free* $\Rightarrow \forall S_i, S_j \ S_i \cap S_j = \emptyset$ or $S_i = S_j$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \neq S_2$ and $S_1 \cap S_2 \neq \emptyset$. We have two cases:

1. $S_1 \subset S_2$ or $S_2 \subset S_1$

By Lemma 3.7, G has a dart, contradiction.

2. $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$

By Lemma 3.8, G has a gem or a butterfly, contradiction.

By Remark 3.10, this is equivalent to say that G is *(dart, gem)-free.*

Therefore $S_i \cap S_j = \emptyset$ or $S_i = S_j \Leftrightarrow G$ is *(dart, gem)-free.*

- (iv) $\forall S_i, S_j$ $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \subset S_j$ or $S_j \subset S_i) \Leftrightarrow G$ is *(gem, butterfly)-free.*

($\Rightarrow$) $\forall S_i, S_j$ $S_i = S_j$, $S_i \cap S_j = \emptyset$ or $(S_i \subset S_j$ or $S_j \subset S_i) \Rightarrow G$ is (gem, butterfly)-free.
If G has an induced gem then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \neq S_2$, $S_1 \cap S_2 \neq \emptyset$ and $S_1 \nsubseteq S_2$ and $S_2 \nsubseteq S_1$, contradiction. Analogously if it has a butterfly.

($\Leftarrow$) G is $(gem, butterfly)$-free $\Rightarrow \forall S_i, S_j$, $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \subset S_j$ or $S_j \subset S_i)$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \neq S_2$, $S_1 \cap S_2 \neq \emptyset$ and $S_1 \not\subset S_2$ and $S_2 \not\subset S_1$. We have only one case:

$S_1 \nsubseteq S_2$ and $S_1 \nsubseteq S_2$ (overlap)
By Lemma 3.8, G has a gem or a butterfly, contradiction.

Therefore $\forall S_i, S_j$ $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \subset S_j$ or $S_j \subset S_i) \Leftrightarrow G$ is *$(gem, butterfly)$-free.*

- (v) $\forall S_i, S_j$ $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \nsubseteq S_j$ and $S_j \nsubseteq S_i) \Leftrightarrow G$ is *dart-free.*

($\Rightarrow$) $\forall S_i, S_j$ $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \nsubseteq S_j$ and $S_j \nsubseteq S_i) \Rightarrow G$ is *dart-free.*
If G has an induced dart then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \subset S_2$, contradiction.

($\Leftarrow$) G is *dart-free* $\Rightarrow \forall S_i, S_j$, $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \nsubseteq S_j$ and $S_j \nsubseteq S_i)$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \subset S_2$. By Lemma 3.7, G has a dart, contradition.

Therefore $\forall S_i, S_j$ $S_i = S_j$ or $S_i \cap S_j = \emptyset$ or $(S_i \nsubseteq S_j$ and $S_j \nsubseteq S_i) \Leftrightarrow G$ is *dart-free.*

- (vi) $\forall S_i, S_j$ $S_i = S_j$ or $(S_i \subset S_j$ or $S_j \subset S_i) \Leftrightarrow G$ is *$(P_4, 2P_3)$-free.*

($\Rightarrow$) $\forall S_i, S_j$ $S_i = S_j$ or $(S_i \subset S_j$ or $S_j \subset S_i) \Rightarrow G$ is $(P_4, 2P_3)$*-free.*
If G has an induced P_4 then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 \neq S_2$ and $S_1 \nsubseteq S_2$ and $S_2 \nsubseteq S_1$, contradiction. Analogously if it has a $2P_3$.

($\Leftarrow$) G is $(P_4, 2P_3)$*-free* $\Rightarrow \forall S_i, S_j$, $S_i = S_j$ or $(S_i \subset S_j$ or $S_j \subset S_i)$.
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \neq S_2$, $S_1 \not\subset S_2$ and $S_2 \not\subset S_1$. We have two cases:

1. $S_1 \cap S_2 = \emptyset$

By Lemma 3.5, G has a P_4 or a $2P_3$, contradiction.

2. ($S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$) (overlap)
By Lemma 3.8, G has a gem or a butterfly; by Remark 3.10, G has a P_4 or a $2P_3$, contradiction.

Therefore $\forall S_i, S_j$ $S_i = S_j$ or ($S_i \subset S_j$ or $S_j \subset S_i$) $\Leftrightarrow$ G is *($P_4, 2P_3$)-free.*

- (vii) $\forall S_i, S_j$, $S_i \cap S_j = \emptyset$ or ($S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$) $\Leftrightarrow$ G is *claw-free*

($\Rightarrow$) $\forall S_i, S_j$ $S_i \cap S_j = \emptyset$ or ($S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$) $\Rightarrow$ G is claw-free.
If G has an induced claw then it has a pair of minimal vertex separators S_1, S_2 such that $S_1 = S_2$; hence $S_1 \cap S_2 \neq \emptyset$ and they do not overlap, contradiction.

($\Leftarrow$) G is claw-free $\Rightarrow$ $\forall S_i, S_j$, $S_i \cap S_j = \emptyset$ or ($S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$).
Now suppose that there exists a pair of minimal vertex separators S_1, S_2 in G such that $S_1 \cap S_2 \neq \emptyset$ and they do not overlap. We have only two cases:

1. $S_1 = S_2$
By Lemma 3.6, G has a claw, contradiction.

2. $S_1 \subset S_2$ or $S_2 \subset S_1$
By Lemma 3.7, G has a dart, contradiction.

Hence by Remark 3.10 G is *claw-free.*

Therefore $\forall S_i, S_j$ $S_i \cap S_j = \emptyset$ or ($S_i \not\subseteq S_j$ and $S_j \not\subseteq S_i$) $\Leftrightarrow$ G is *claw-free.* □

We remark that *(iii)* had been previously proved in [43], [17] and [55] and they are exactly the strictly chordal graphs.

The previous theorem deals with unique possible cases (non-trivial) of combinations. In fact, as we have four situations, we have sixteen possibilities. Of these, seven are interesting, two are trivial (when all situations can happen or anyone can happen), and by Remark 3.9 seven of them are impossible, as shown in Tables 3.1 and 3.2.

	Disjunction	Equality	Containment	Overlap	Forbidden subgraphs
1 (i)	x				(claw,gem)
2 (iii)	x	x			(dart,gem)
3 (ii)		x			(P_4, gem, butterfly)
4 (vi)		x	x		($P_4, 2P_3$)
5 (iv)	x	x	x		(gem, butterfly)
6 (v)	x	x		x	dart
7(vii)	x			x	claw
8	x	x	x	x	trivial
9					trivial

Table 3.1: Possible cases

	(a) Disjunction	(b) Equality	(c) Containment	(d) Overlap	Impossible
10	x		x		$c \Rightarrow b$
11	x		x	x	$c \Rightarrow b$
12		x		x	$d \Rightarrow a$
13		x	x	x	$d \Rightarrow a$
14			x		$c \Rightarrow b$
15			x	x	$c \Rightarrow b$, $d \Rightarrow a$
16				x	$d \Rightarrow a$

Table 3.2: Impossible cases

3.3 Helly Property

We now move our attention to the study of the Helly property. In 1923 Eduard Helly published his famous work [36] with a result about convexity of a family of sets and it originated what is now known as Helly property, which has been extensively studied in several branches of mathematics and other areas. In graph theory, Helly property arises in several works, such as [1, 20, 27, 65]. In particular we note that this property has a strong relationship with the structure of chordal graphs and the vertices of the clique tree, since subtrees of a tree satisfy the Helly property [30].

Let $\mathcal{F}$ be a family of subsets of a set $\mathcal{S}$. We say that $\mathcal{F}$ satisfies the Helly property when every subfamily $\mathcal{F}'$ of $\mathcal{F}$ consisting of pairwise intersecting subsets satisfies $\bigcap_{F \in \mathcal{F}'} F \neq \emptyset$.

Example 3.12. *On the following we have a family satisfying Helly property (a) and other family not satisfying Helly property (b).*

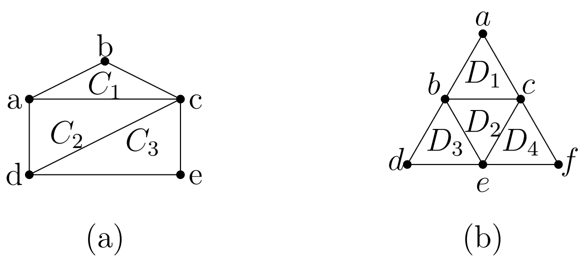


Figure 3.10: (a) $C_1 = \{a, b, c\}$, $C_2 = \{a, c, d\}$, $C_3 = \{c, d, e\}$ (b) $D_1 = \{a, b, c\}$, $D_2 = \{b, c, e\}$, $D_3 = \{b, d, e\}$, $D_4 = \{c, e, f\}$

For the graph of Figure 3.10 (a), if we consider the family of cliques $\mathcal{F} = \{C_1, C_2, C_3\}$, as we have $C_1 \cap C_2 \cap C_3 = \{c\}$, then for every subfamily $\mathcal{F}'$ of $\mathcal{F}$ we have $\cap_{C_i \in \mathcal{F}'} C_i \neq \emptyset$ and then Helly property is satisfied. On the graph of Figure 3.10 (b) if we take family the $\mathcal{F} = \{C_1, C_2, C_3, C_4\}$, we have the subfamily $\mathcal{F}' = \mathcal{F}$ consisting of pairwise intersecting subsets C_i but $\cap_{C_i \in \mathcal{F}'} C_i = \emptyset$ and then it does not satisfy Helly property.

For the next results we use a family as witness. We say that a family X is a *witness* that the Helly property does not hold if:

- $S_i \cap S_j \neq \emptyset$, $\forall S_i, S_j \in X$ and
- $\bigcap_{S_i \in X} S_i = \emptyset$.

Lemma 3.13. *Let G be a chordal graph such that $\mathbf{S}(G)$ does not satisfy the Helly property and such that for every induced subgraph G' the family $\mathbf{S}(G')$ satisfies the Helly property. Let $\mathcal{T}$ be a clique tree of G and $\mathbf{S}_\ell(T)$ be the set of minimal vertex separators incident to leaves of $\mathcal{T}$. Then $\mathbf{S}_\ell(T)$ is a witness.*

Proof. If there exist separators $S_i, S_j \in \mathbf{S}_\ell(T)$ with $S_i \cap S_j = \emptyset$ then S_i, S_j cannot be simultaneously in a witness. But then there exists witness $R \subset \mathbf{S}(G)\backslash S_i$ or $R \subset \mathbf{S}(G)\backslash S_j$, which contradicts the minimality of G. Then suppose that $S_i \cap S_j \neq \emptyset$ for all pairs $S_i, S_j \in \mathbf{S}_\ell(T)$. Since $\mathbf{S}_\ell(T)$ is not a witness there exists $x \in \bigcap_{S_i \in \mathbf{S}_\ell(T)} S_i$ and then x is universal in G and this implies that x belongs to all minimal vertex separators of G. But this implies that $\mathbf{S}(G)$ satisfies the Helly property, a contradiction. $\square$

Theorem 3.14. *Let $\mathcal{G}$ be the hereditary class of chordal graphs such that $\forall G \in \mathcal{G}$, $\mathbf{S}(G)$ satisfies the Helly property. Then for any chordal graph G, $G \in \mathcal{G} \Leftrightarrow G$ is* Hajós-free.

Proof. Let $G \in \mathcal{G}$ and let $\{x, y, z, a, b, c\}$ be an induced Hajós of G, with cliques $C_1 = \{x, y, a\}, C_2 = \{x, z, b\}, C_3 = \{y, z, c\}, C_4 = \{x, y, z\}$. Let $S_1 = C_1 \cap C_4 = \{x, y\}, S_2 = C_2 \cap C_4 = \{x, z\}, S_3 = C_3 \cap C_4 = \{y, z\}$, see Figure 3.11. Then we have $S_1 \cap S_2 = \{x\}, S_1 \cap S_3 = \{y\}, S_2 \cap S_3 = \{z\}$ and $S_1 \cap S_2 \cap S_3 = \emptyset$, so $\mathbf{S}(G)$ does not satisfy the Helly property.

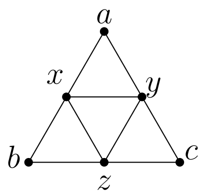


Figure 3.11: Hajós

Conversely let $G \notin \mathcal{G}$ and let G' be an induced subgraph of G minimal in relation to the property that $\mathbf{S}(G')$ does not satisfy the Helly property and let $\mathcal{T}'$ be a clique tree of G'. By the previous Lemma, $\mathbf{S}_\ell(T')$ is a witness. Let $S_1, S_2 \in \mathbf{S}_\ell(T')$. Then $\exists x_1 \in S_1 \cap S_2$ and $\exists S_i \in \mathbf{S}_\ell(T')$ such that $x_1 \notin S_i$. Without lost of generality, let $S_i = S_3$. Note that if $A, B, C \in \mathbf{S}_\ell(T')$ then $A \cap B \nsubseteq C$. Indeed, suppose $A \cap B \subseteq C$. Since $\mathbf{S}_\ell(T')$ is a witness, we know that $\bigcap_{X \in \mathbf{S}_\ell(T')} X = \emptyset$. Then we have:

$$\left(\bigcap_{X \in \mathbf{S}_\ell(T') \backslash \{A,B,C\}} X \right) \cap A \cap B \cap C = \emptyset \Rightarrow$$

$$\left(\bigcap_{X\in \mathbf{S}_\ell(T')\setminus\{A,B,C\}} X\right)\cap A\cap B=\emptyset \Rightarrow$$
$$\bigcap_{X\in \mathbf{S}_\ell(T')\setminus\{C\}} X=\emptyset.$$

But this implies that there exists a proper subset of $\mathbf{S}_\ell(T')$ that does not satisfy the Helly property, contradicting the minimality of G'. Then we can suppose that $\exists y \in S_3 \cap S_1\setminus S_2$ and $\exists z \in S_3 \cap S_2 \setminus S_1$. Let C_1, C_2, C_3 be the leaves of edges labeled by S_1, S_2, S_3 respectively in $\mathcal{T}'$. Let v_i be an exclusive vertex of $C_i, i = 1\ldots 3$. Then $\{x, y, z, v_1, v_2, v_3\}$ induces a Hajós. $\square$

Chapter 4

Reconfiguration

4.1 Introduction

The idea of reconfiguration in graphs seems quite natural: given two sets of vertices with a given property, we wish to know if it is possible to turn one set into another, in a sequence that at each step one vertex is changed, so that the intermediate sets in each step remain the same property.

Reconfiguration Problems consist of a transformation step by step from a particular solution S_a of a problem instance to another particular solution S_b such that all intermediate steps in the sequence $S_a = S_1, S_2, \ldots, S_n = S_b$ are also feasible solutions of the problem. In order to deal with reconfiguration, we must have a definition of feasible solution and adjacency of feasible solutions. In [61] we have an important survey about reconfiguration in graphs, that discusses techniques, results and possible directions of research in this area.

A very useful example to think about reconfiguration is the well known 15-puzzle, see Figure 4.1. It consists of a 4×4 box with 15 squares numbered from 1 to 15 and one square empty. The goal of the puzzle is, given an arbitrary starting arrangement of the numbers, as for example Figure 4.1(a), to obtain the configuration shown in Figure 4.1(b) where the numbers are ordered and the last square is empty. In order move from one configuration to the next one, one can slide a number to the empty position, if that number is in a neighboring cell. This game was studied in [40] in 19th century, as well as the complexity of other games was studied in recent works, such as Rubik's cube [42].

When modeled on a graph, the problem has the shape of the Figure 4.2. Note that this game is just a simplified illustration of the reconfiguration problem. In it, we are not interested in the intermediate steps between initial and final configuration and

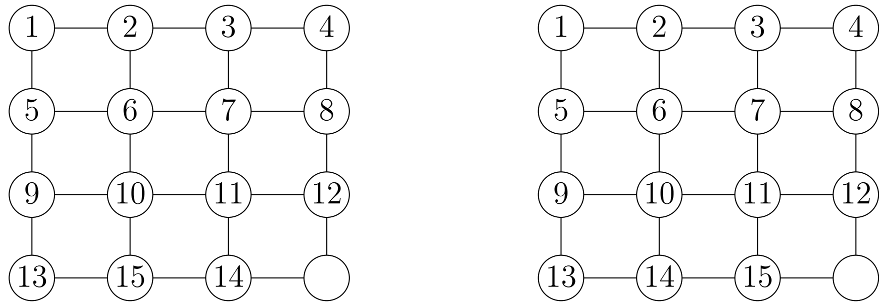
(a)

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

(b)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Figure 4.1: 15 puzzle (a) Initial configuration (b) Final Configuration



(a) (b)

Figure 4.2: (a) Initial configuration (b) Final configuration

also the only rule for label change of a vertex is that one can slide the label of a vertex to the blank vertex, as long as they are neighbors. In general, for the most interesting graph reconfiguration problems, we make more demands on intermediate solutions as well as reconfiguration rules.

Graph Theory had its beginning with the work known with the Seven Bridges of Königsberg, by Leonard Euler in 1736 [21], which is also considered to be a work that launched the first ideas of Topology; this is an example of a problem linking different areas of mathematics, in this case Topology and Graph Theory.

A very common example to show the object of study in topology is to say that the torus (or a donut) and a mug are topologically indistinguishable, or homeomorphic, that is, we can transform each other only by deformations (stretching, kneading, bending, and so on) without, however, breaking (cutting or pasting) them.

Many real-world problems look like this: given the description of a system state and the description of a desired one, we wonder if it is possible to transform the system from its current state into the desired one without breaking the system. Of course in every situation you need to be clear what it means to keep the system without breaking.

Recently reconfiguration problems have arisen from computational problems in different areas such as Graph Theory [13, 37], Satisfiability [32], Computational Geometry [18], Quantum Complexity Theory [25].

In the context of Graph Reconfiguration, the problem can be posed as follows: Given two sets of vertices of a graph G that have a common property P, it is possible to transform one set into another by a sequence of steps such that at each step only a vertex is changed so that all sets obtained in each step preserve the property P?

Graph Reconfiguration problems has been studied extensively for several well-known problems, considering different important structures, such as Independent Set [8, 37, 41], Shortest Paths [9], Vertex Cover [39, 60], Clique [38], Matching [37], Vertex Coloring [10].

4.2 Complexity

The complexity of Graph Reconfiguration problems has been studied extensively for several well-known problems, considering different important structures. Kamiński et al. [41] (2012) proved that Independent Set Reconfiguration problem is a PSPACE-complete problem for perfect graphs and they gave polynomial-time results for even-hole-*free* graphs and P_4-*free* graphs; Bonsma and Cereceda [9] (2013) proved that Shortest Paths Reconfiguration problem is PSPACE-complete for general graphs and it can be solved in polynomial time for *claw-free* graphs and chordal graphs; Ito et al. [39] (2016) proved that Vertex Cover Reconfiguration problem is PSPACE-complete for planar graphs and they gave a linear-time algorithm to solve the problem for even-hole-*free* graphs, which include several well-known graphs, such as trees, interval graphs and chordal graphs; Ito et al. [38] (2011) proved that Clique Reconfiguration problem is PSPACE-complete for perfect graphs and they gave polynomial-time algorithms for several classes of graphs, such as even-hole-*free* graphs and cographs; Ito at al. [37] proved that Matching Reconfiguration problem can be solved in polynomial time for general graph; Bonsma and Cereceda[10] (2009) proved that finding paths between graph k-colorings is PSPACE-complete for $k \geq 4$ for bipartite graphs and Cereceda et al [12] (2011) proved that given a 3-colorable graph proper vertex 3-colorings reconfiguration can be solved in polynomial time.

When considering reconfiguration versions of classic problems, sometimes surprisingly different results arise. In general, simple combinatorial problems often give rise to tractable reconfiguration problems. But there exist examples that contradict this pattern. While we know that finding a shortest $s - t$ path in graph is computationally easy, finding a sequence of transformation steps between two shortest paths is PSPACE-complete [8]. On the other hand we have the 3-coloring problem: finding a solution is NP-complete, and finding a reconfiguration sequence between two given solutions can be done in polynomial time [12].

4.3 VERTEX SEPARATOR RECONFIGURATION in graphs

Within the scope of this work, we will deal with complexity of some VERTEX SEPARATOR RECONFIGURATION problems. Given a graph G, two vertices u, v of G, and two vertices sets S_a and S_b of G that separate u and v, we are asked: is there a a sequence of steps $S_a = S_0, S_1, S_2, \ldots, S_n = S_b$ such that each set $S_i, i = 1, \ldots, n$ also separates u and v in G and it is obtained from the previous one by changing of one vertex following a certain specified rule?

As mentioned before, in Reconfiguration problems two concepts are necessary: feasibility and adjacency. In our problem, given a graph G we fix two vertices in G and two vertex separator sets of these vertices; a feasible solution is a subset of vertices of G that separates the fixed vertices. Two solutions are adjacent if one can be transformed into another by a single reconfiguration step according to some rule, or equivalently, we can think of the placement of a set of tokens on a subset of vertices of G and the reconfiguration steps are a change in the placement of the tokens. And in this work we study the well known reconfiguration rules Token sliding, Token Jumping and Token Addition/Removal, considered in many works as in [35, 37, 41] and others.

In Token Sliding (TS), first introduced by [35], a step can be understood as to slide a token along an edge and two solutions are adjacent if:

- (i) they have the same cardinality,
- (ii) the size of their intersection is one less than the size of the solutions and
- (iii) the two vertices outside the intersection are adjacent.

In Token Jumping (TJ), first introduced by [41], we remove the constraint that the vertices outside the intersection are adjacent, and then a step can be understood as a token jumping from a vertex to any other vertex and two solutions are adjacent if they satisfy conditions (i) and (ii).

In Token Addition/Removal (TAR), first introduced by [37], we allow a token to be either added or removed at each reconfiguration step and the sizes of two solutions will differ by exactly one token.

Denoting the adjacency relation by $\leftrightarrow$, we can define formally the previous relations as following:

Let S_i, S_j be two separator sets in G. We consider the adjacency relations:

- **TS**: $S_i \leftrightarrow S_j$ if $|S_i| = |S_j|$, $S_i \setminus S_j = \{u_i\}$, $S_j \setminus S_i = \{v_j\}$ and $u_iv_j \in E(G)$.
- **TJ**: $S_i \leftrightarrow S_j$ if $|S_i| = |S_j|$ and $|S_i \setminus S_j| = |S_j \setminus S_i| = 1$
- **TAR**: $S_i \leftrightarrow S_j$ if $|S_i \Delta S_j| = |(S_i \setminus S_j) \cup (S_j \setminus S_i)| = 1$

In fact, for many situations a simple *TAR* sequence without a threshold k for each set in the sequence makes the problem trivial, because we can simply add or remove vertices. To keep the question interesting and avoid this triviality phenomenon, a lower or upper bound is imposed on the cardinality of the intermediate token set. For example: if we want to reconfigure a clique C_1 in a clique C_2, in a graph G in a way that every intermediate step is also a clique, we can remove vertices from C_1 until obtain the empty set and after this we can add vertices of C_2, one by one, until obtain C_2. So, in this case, the problem is only of interest if we define a lower limit k for cliques on each step. On the other hand, if we want to reconfigure an independent set I_1 in an independent set I_2, in a graph G in a way that every intermediate step is also an independent set, we can add vertices to I_1 until obtain the set $I_1 \cup I_2$ and after this we can remove vertices of $I_1 cup I_2$, one by one, until obtain I_2. So, in this case, the problem is only of interest if we define a upper limit k for independent sets on each step.

Analogously for Vertex Separators Sets Reconfiguration: the problem is only of interest if we define a upper limit k for separators in the sequence; otherwise, given two vertex separators S_a and S_b, if we desire reconfigure S_a into S_b we could simply add the vertices of S_b to S_a until we get $S_a \cup S_b$ and then remove vertices of $S_a \cup S_b$ until we get S_b. Then in our case, we will add the restriction that the intermediate vertex separator sets s must have at most k vertices. Therefore the threshold k will be clear in the context of the problem. Since the direction of the bound is usually clear by the problem definition, the bounded version of TAR is usually referred to as *TAR(k)*. Throughout our text, unless the bound k is relevant to the discussion, in an abuse of terminology, we use simply *TAR*.

In the context of vertex separators reconfiguration from a set S_a to a set S_b of u, v-vertex separators for a given pair of vertices u and v of a graph G we write $S_a \leftrightsquigarrow S_b$ by TAR if there exists a sequence $\langle S_a = S_0, S_1, \ldots, S_n = S_b \rangle$ of u, v-vertex separators in G such that each S_i is obtained from S_{i-1}, $i = 1, 2, \ldots, n$, by addition or removal of a vertex $v_i \in V(G)$. And we note that

$$TAR(G, S_a, S_b) = \begin{cases} Yes, & \text{if } S_a \leftrightsquigarrow S_b \text{ by } TAR, \\ No, & \text{otherwise.} \end{cases}$$

Analogous functions can be defined to TJ and TS.

Given a TAR-instance (G, S_a, S_b, k), let $\mathbf{S} = \langle S_0, S_1, \ldots, S_\ell \rangle$ be a TAR(k)-sequence in G between $S_a = S_0$ and $S_b = S_\ell$. The length of $\mathbf{S}$ is defined as the number of sets in $\mathbf{S}$ minus one, that is, the length of $\mathbf{S}$ is ℓ. We denote by $dist_{TAR}(G, S_a, S_b, k)$ the minimum length of a $TAR(k)$-sequence in G between S_a and S_b; we let $dist_{TAR}(G, S_a, S_b, k) = +\infty$ if there is no $TAR(k)$-sequence in G between S_a and S_b. Similarly we define $dist_{TJ}(G, S_a, S_b)$ and $dist_{TS}(G, S_a, S_b)$ for a TJ and a TS-instance (G, S_a, S_b), respectively.

Our main results in this chapter are:

- TJ and TAR are equivalent in Vertex Separator Reconfiguration in a precise sense;
- Vertex Separator Reconfiguration is PSPACE-hard under TS and NP-hard under TAR/TJ for bipartite graphs;
- Vertex Separator Reconfiguration problem can be solved in polynomial time under TAR/TJ for graphs with polynomially bounded number of minimal vertex separators;
- We show a graph class that does not have a polynomially bounded number of minimal vertex separators, but Vertex Separator Reconfiguration problem can be solve in polynomial time, showing that a superpolynomial number of minimal vertex separators does not imply hardness.

Graph class	**TAR**	**TJ**	**TS**
Bipartite	NP-hard	NP-hard	PSPACE-hard
$\{3P_1, diamond\}-free$	Polynomial	Polynomial	Polynomial
Polynomially bounded number of minimal vertex separators	Polynomial	Polynomial	Open

Table 4.1: Results in Vertex Separator Reconfiguration

Table 4.1 summarizes complexity results for Vertex Reconfiguration problems that we studied in this work.

4.4 TAR/TJ Equivalence

Ito et al. [38] studied Clique Reconfiguration, which consists of transforming a clique in another one by the rules TS, TJ and TAR and they proved that all the three rules are equivalent in the sense of complexity, as shown in Theorems 4.1 and 4.2.

Theorem 4.1 ([38]). *TS and TAR rules are equivalent in CLIQUE RECONFIGURATION, as follows:*

- *(a) for any TS-instance (G, C_0, C_r), a TAR-instance (G, C'_0, C'_r, k') can be constructed in linear time such that $TS(C_0, C_r) = TAR(C'_0, C'_r, k')$ and $dist_{TS}(C_0, C_r) = dist_{TAR}(C'_0, C'_r, k')/2$ and*
- *(b) for any TAR-instance (G, C_0, C_r, k), a TS-instance (G, C'_0, C'_r) can be constructed in linear time such that $TAR(C_0, C_r, k) = TS(C'_0, C'_r)$.*

Theorem 4.2 ([38]). *TJ and TAR rules are equivalent in CLIQUE RECONFIGURATION, as follows:*

- *(a) for any TJ-instance (G, C_0, C_r), a TAR-instance (G, C'_0, C'_r, k') can be constructed in linear time such that $TJ(C_0, C_r) = TAR(C'_0, C'_r, k')$ and $dist_{TJ}(C_0, C_r) = dist_{TAR}(C'_0, C'_r, k')/2$ and*
- *(b) for any TAR-instance (G, C_0, C_r, k), a TJ-instance (G, C'_0, C'_r) can be constructed in linear time such that $TAR(C_0, C_r, k) = TJ(C'_0, C'_r)$.*

In a similar fashion we now prove that *TJ* and *TAR* are equivalent in VERTEX SEPARATOR RECONFIGURATION, in the sense that, given a graph G and two vertex separators S_a, S_b in G, if we have a *TJ*-instance (G, S_a, S_b) we can construct a $TAR-$instance (G, S'_a, S'_b, k) such that $TJ(G, S_a, S_b) = TAR(G, S'_a, S'_b, k)$ and if we have a *TAR*-instance (G, S_a, S_b, K) we can construct a $TJ-$instance (G, S'_a, S'_b) such that $TAR(G, S_a, S_b, K) = TJ(G, S'_a, S'_b)$. Our proofs are inspired by Ito et al. [38] for Clique Reconfiguration.

First we prove a lemma that gives us that if S_a, S_b are two uv-vertex separators such that $|S_a| = |S_b|$ and $S_a \leftrightsquigarrow S_b$ by TAR, there exists a shortest TAR sequence that can be constructed adding and removing a vertex in each step.

Lemma 4.3. *Let G be a graph, $u, v \in V(G)$ and let S, S' be a pair of Vertex separators (of u, v) in G such that $|S| = |S'| = k$ and $S \leftrightsquigarrow S'$ under $TAR(k+1)$. Then there exists a shortest $TAR(k+1)$-sequence $\langle S_0, S_1, \ldots, S_\ell \rangle$ from $S_0 = S$ to $S_\ell = S'$ such that $|S_{2i-1}| = k+1$ e $|S_{2i}| = k$, $\forall i \in \{1, 2, \ldots, \ell/2\}$.*

Proof. Let $\mathbf{S} = \langle S_0, S_1, \ldots, S_\ell \rangle$ be a shortest $TAR(k+1)$ sequence from $S_0 = S$ to $S_\ell = S'$ that maximizes the sum $\sum_{i=0}^{l} |S_i|$. Since each separator in the $TAR(k+1)$-sequence is of size at most $k+1$, it suffices to show that $|S_j| \geq k$, for $j \in \{1, 2, \ldots, l-1\}$.

Let s be an index satisfying $|S_s| = \min_{i=0}^{l} |S_i|$ and suppose by contradiction that $|S_s| \leq k-1$. By definition of s, we have that $S_{s-1} \supset S_s \subset S_{s+1}$. Let a, b vertices such that $S_s = S_{s-1} \setminus \{a\}$ and $S_{s+1} = S_s \cup \{b\} = (S_{s-1} \setminus \{a\}) \cup \{b\}$. Note that since $\langle S_0, S_1, \ldots, S_l \rangle$ is shortest, we have $a \neq b$ and then $b \notin S_{s-1}$. We now replace the separator S_s by another $S'_s = S_{s-1} \cup \{b\}$ and obtain the following sequence

$$\mathbf{S}' = \langle S_0, S_1, \ldots, S_{s-1}, S'_s = S_{s-1} \cup \{b\}, S_{s+1}, \ldots, S_l \rangle.$$

Since $S_{s-1} = S_s \cup \{a\}$ and $|S_s| \leq k-1$ we have $|S'_s| = |S_s \cup \{a, b\}| \leq k+1$ and then $S_{s-1} \leftrightarrow S_{s-1} \cup \{b\} = S'_s$ under $TAR(k+1)$. Furthermore $S_{s+1} = (S_{s-1} \setminus \{a\}) \cup \{b\} = S'_s \setminus \{a\}$. Then we have $S'_s \leftrightarrow S_{s+1}$ under $TAR(k+1)$. Therefore $\mathbf{S}'$ is a $TAR(k+1)$-sequence between S and S'.

Note that $\mathbf{S}'$ is of length ℓ and hence it is a shortest $TAR(k+1)$- sequence between S and S'. Since $S'_s = S_s \cup \{a, b\}$ we have $|S'_s| > |S_s|$ and hence

$$|S'_s| + \sum \{|S_j|; j \in \{0, 1, \ldots, s-1, s+1, \ldots, l\}\} > \sum_{i=0}^{l} |S_i|.$$

This contradicts the assumption that $\mathbf{S} = \langle S_0, S_1, \ldots, S_l \rangle$ is a shortest $TAR(k+1)$-sequence from $S_0 = S$ to $S_l = S'$ that maximizes the sum $\sum_{i=0}^{l} |S_i|$. □

Now we prove the following Lemma:

Lemma 4.4. *Let G be a graph and let S_0, S_r be any pair of vertex separators in G such that $|S_0| = |S_r| = k$. Then $TJ(G, S_0, S_r) = TAR(G, S_0, S_r, k+1)$ and $dist_{TJ}(G, S_0, S_r) = (dist_{TAR}(G, S_0, S_r, k+1))/2$.*

Proof. We first prove that if $TJ(G, S_0, S_r) =$ YES then $TAR(G, S_0, S_r, k+1) =$ YES. Suppose that there exists (G, S_0, S_r) a *TJ*-instance such that $TJ(G, S_0, S_r) =$YES. Then there exists a *TJ* sequence between S_0 and S_r; let $\mathbf{S} = \langle S_0, S_1, \ldots, S_l \rangle$ be a shortest sequence, that is, $S_l = S_r$ and $l = dist_{TJ}(G, S_0, S_r)$. For each $j \in \{1, 2, \ldots, l\}$, let $S_{j-1} \setminus S_j = \{u_j\}$ and $S_j \setminus S_{j-1} = \{w_j\}$. Now for each S_j of the sequence, we replace it by $\langle S_j \cup \{w_{j+1}\}, S_j \rangle$ and we obtain the following sequence of vertex separators:

$$\mathbf{S}' = \langle S_0, S_0 \cup \{w_1\}, S_1, S_1 \cup \{w_2\}, \ldots, S_{l-1} \cup \{w_l\}, S_l \rangle.$$

Note that $S_j \cup \{w_{j+1}\} \leftrightarrow S_{j+1}$ under $TAR(k+1)$ for each $j \in \{0,1,2,\dots,l-1\}$, because $S_{j+1} = (S_j \cup \{w_{j+1}\}) \setminus \{u_j\}$. Hence the sequence $\mathbf{S}'$ above is a $TAR(k+1)$-sequence from S_0 to S_r and hence $TAR(G, S_0, S_r, k+1) =$ YES. Furthermore, by construction, $\mathbf{S}'$ is of length $2l$. Hence

$$dist_{TAR}(G, S_0, S_r, k+1) \leq 2l = 2 \cdot dist_{TJ}(G, S_0, S_j). \tag{4.1}$$

We prove now that if $TAR(G, S_0, S_r, k+1) =$ YES then $TJ(G, S_0, S_r) =$ YES. Suppose then that there exists a $TAR(k+1)$-sequence from S_0 to S_r. Let $\mathbf{S} = \langle S_0, S_1, \dots, S_m \rangle$ be a shortest sequence, that is, $S_m = S_r$ and $m = dist_{TAR}(G, S_0, S_r, k+1)$. Furthermore, by Lemma 4.3, we can assume that $|S_{2i-1}| = k+1$ and $|S_{2i}| = k, \forall i \in \{1,2,\dots,m\}$. Let $S_{2i-1} = S_{2i-2} \cup \{u_{2i-2}\}$ and $S_{2i} = S_{2i-1} \setminus \{w_{2i}\}$. Since $\mathbf{S} = \langle S_0, S_1, \dots, S_m \rangle$ is shortest, we have $u_{2i-2} \neq w_{2i}$. Then for every $i \in \{1,2,\dots,m\}$ we have that $S_{2i-2} \leftrightarrow S_{2i}$ under TJ. We now replace each pair $\langle S_{2i-1}, S_{2i} \rangle$ by S_{2i}, obtaining the sequence

$$\mathbf{S}'' = \langle S_0, S_2, S_4, \dots, S_m \rangle.$$

In this way, S'' is a TJ-sequence from S_0 to S_m and therefore $TJ(G, S_0, S_r) =$ YES. Moreover, S'' is of length $m/2$. Hence

$$dist_{TJ}(G, S_0, S_r) \leq m/2 = \left(dist_{TAR}(G, S_0, S_r, k+1)\right)/2. \tag{4.2}$$

By (4.1) and (4.2), we have that

$$dist_{TJ}(G, S_0, S_r) = \left(dist_{TAR}(G, S_0, S_r, k+1)\right)/2 \qquad \square$$

.

Lemma 4.5. *Let (G, S_0, S_r, k) be a $TAR(k)$-instance such that $S_0 \neq S_r$. Suppose that there exists an index $j \in \{0, r\}$ such that $|S_j| = k$ and S_j is a minimal vertex separator in G. Then $TAR(G, S_0, S_r, k) =$* NO.

Proof. Since S_j is minimal, does not exist vertex separator in G that can be obtained by removing a vertex from S_j. Furthermore, since $|S_j| = k$, we cannot add a vertex to S_j to keep the threshold k. Then there is no vertex separator S in G such that $S_j \leftrightarrow S$ under $TAR(k)$. Since $S_0 \neq S_r$, we have $TAR(G, S_0, S_r, k) =$NO. $\square$

Theorem 4.6. *Let G be a graph and let u, v be two vertices of G and S_a, S_b two uv-vertex separators. Then TJ and TAR are equivalent in* VERTEX SEPARATOR RECONFIGURATION *in the following sense:*

- *(a) If (G, S_a, S_b) is a TJ-instance then we can construct in linear time a TAR-instance (G, S'_a, S'_b, k') such that $TJ(G, S_a, S_b) = TAR(G, S'_a, S'_b, k')$ and $dist_{TJ}(G, S_a, S_b) = (dist_{TAR}(G, S'_a, S'_b, k'))\,/2$;*

- *(b) If (G, S_a, S_b, k) is a TAR-instance then we can construct in linear time a TJ-instance (G, S'_a, S'_b) such that $TJ(G, S'_a, S'_b) = TAR(G, S_a, S_b, k)$.*

Proof.

- (a) Let (G, S_a, S_b) be a TJ-instance with $|S_a| = |S_b| = k$. Then as the corresponding TAR-instance (G, S'_a, S'_b, k') we let $S'_a = S_a$, $S'_b = S_b$ and $k' = k+1$. Clearly this TAR-instance can be constructed in linear time. By Lemma 4.4, we have that
$$TJ(G, S_a, S_b) = TAR(G, S'_a, S'_b, k')$$
and
$$dist_{TJ}(G, S_a, S_b) = (dist_{TAR}(G, S'_a, S'_b, k'))\,/2.$$
This completes the proof of Theorem (a).

- (b) Let (G, S_a, S_b, k) be a $TAR(k)$-instance; $|S_a| \neq |S_b|$ may hold and both $|S_a| \leq k$ and $|S_b| \leq k$ must hold. By Lemma 4.5, we can assume without loss of generality that none S_a and S_b is a minimal separator in G of size k. Then we construct a corresponding TJ-instance (G, S'_a, S'_b) as follows:

- (i) for each $j \in \{a, b\}$ such that $|S_j| \leq k-1$ let $S'_j \supseteq S_j$ be an arbitrary superset of size $k-1$.

- (ii) for each $j \in \{a, b\}$ such that $|S_j| = k$ let $S'_j \subseteq S_j$ be an arbitrary subset of size $k-1$.

Clearly, this TJ-instance can be constructed in linear time. We then prove the following Lemma:

Lemma 4.7. *Let (G, S_a, S_b, k) be a TAR-instance and let (G, S'_a, S'_b) be the corresponding TJ-instance constructed above in (i) and (ii). Then $TAR(G, S_a, S_b, k) = TJ(G, S'_a, S'_b)$.*

Proof. Given a TAR-instance (G, S_a, S_b, k)-instance, let (G, S'_a, S'_b) be a TJ-instance constructed as (i) and (ii). Since $|S'_a| = |S'_b| = k-1$, by Lemma 4.4, we have
$$TJ(G, Sa', S'_b) = TAR(G, S'_a, S'_b, k). \tag{4.3}$$

Note that $S'_a \supseteq S_a$ or $S_a \supseteq S'_a$ and then in both cases $S_a \leftrightsquigarrow S'_a$ under $TAR(k)$ adding the vertices of $S'_a \setminus S_a$ to S_a one by one or removing the vertices of $S_a \setminus S'_a$ from

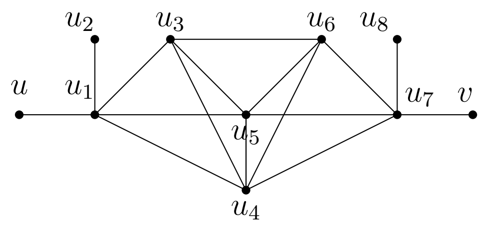


Figure 4.3: Under TAR/TJ, we can reconfigure $S_a = \{u_1, u_2\}$ into $S_b = \{u_7, u_8\}$, but not under TS.

S_a one by one, respectively. Similarly we have $S_b \leftrightsquigarrow S'_b$ under $TAR(k)$. Note that $|S'_a| = |S'_b| = k - 1$.

And now we first prove that if $TJ(G, S'_a, S'_b) =$YES then $TAR(G, S_a, S_b, k) =$YES. In this case, by Eq. (4.3), we have that $TAR(G, S'_a, S'_b, k) =$YES and then $S'_a \leftrightsquigarrow S'_b$ under $TAR(k)$. Hence $S_a \leftrightsquigarrow S'_a \leftrightsquigarrow S'_b \leftrightsquigarrow S_b$ holds under $TAR(k)$ e therefore $TAR(G, S_a, S_b, k) =$ YES.

Finally, we prove that if $TAR(G, S_a, S_b, k) =$ YES, then $TJ(G, S'_a, S'_b) =$YES. In this case, since $TAR(G, S_a, S_b, k) =$YES, we have $S_a \leftrightsquigarrow S_b$ under $TAR(k)$. Hence $S'_a \leftrightsquigarrow S_a \leftrightsquigarrow S_b \leftrightsquigarrow S'_b$ holds under $TAR(k)$ e therefore $TAR(G, S'_a, S'_b, k) =$ YES. By Lemma 4.4, $TJ(G, S'_a, S'_b) =$YES. □

The proof of this emma completes the proof of (b). □

4.4.1 TAR/TS non equivalence

In the sense of equivalence between reconfiguration rules given in Theorem 4.6, we have that *TAR* and *TS* are not equivalent in VERTEX SEPARATOR RECONFIGURATION. In Figure 4.3 we have that $S_a = \{u_1, u_2\}, S_b = \{u_7, u_8\}$ are $uv-$separators in G and if we consider (G, S_a, S_b) a *TS*-instance we can note that $TS(G, S_a, S_b) =$NO and then we cannot transform this *TS*-instance in a $TAR(2)$-instance because every $TAR(2)$-instance is YES in G.

4.5 Hardness results

Now we have our main result about hardness. Lokshtanov and Mouaward [53] proved that Independent Set Reconfiguration problem is PSPACE-complete under *TS* and NP-complete under TAR/TJ for bipartite graphs. In what follows we have a lemma that give us an equivalence between Independent Set and Vertex Separator in bipartite graphs and after this we show a reduction from Bipartite Independent Set Reconfiguration problem to Vertex Separator Reconfiguration problem and conclude that Vertex Separator Reconfiguration is PSPACE-hard under TS and NP-hard under TAR/TJ for bipartite graphs.

Lemma 4.8. *Let G be a bipartite graph with partition A, B. Let H be the graph constructed from G adding two vertices u and v to G such that $N(u) = A$ and $N(v) = B$. Then a set $I \subset V(G)$ is a independent set of G if and only if $V(G) \setminus I$ is a uv-vertex separator in H.*

Proof. Let I be an independent set in G. Suppose by contradiction that $V(G) \setminus I$ is not a uv-vertex separator in H. Then there exists a path $uxyv$ from u to v in H such that $x, y \notin$V(G) $\setminus I$. Hence $x, y \in I$, a contradiction because I is an independent set.

Conversely, let I be a subset of $V(G)$ such that $V(G) \setminus I$ is a uv-Vertex Separator in H. Suppose, to the contrary, that I is not an independent set in G. Let $x, y \in I$ be two adjacent vertices in graph G. Since G is bipartite, we can assume without loss of generality that $x \in A$ and $y \in B$. Then there exists a path $uxyv$ from u to v in H that is does not contain vertices of $V(G) \setminus I$. But this is a contradiction because $V(G) \setminus I$ is a uv-vertex separator in H. □

Theorem 4.9. Vertex Separator Reconfiguration *is NP-hard under TJ for bipartite graphs.*

Proof. We give a polynomial-time reduction from the Bipartite Independent Set Reconfiguration problem to this problem.

Let $I_1 = (G, S, T)$ be an TJ-instance of Bipartite Independent Set Reconfiguration, where G is a bipartite graph with bipartition A, B and S, T are independent sets in G. We can construct $I_2 = (H, S', T')$ a TJ-instance of Vertex Separator Reconfiguration as follows:

- H is obtained from G adding two vertices u, v to G being $N(u) = A$ and $N(v) = B$, as the construction for Lemma 4.8.
- $S' = V(G) \setminus S$ and $T' = V(G) \setminus T$.

It is clear that this TJ-instance can be constructed in linear time.

If $TJ(G, S, T) =$ Yes, let $\langle S = Q_0, Q_1, \ldots, Q_r = T\rangle$ be a sequence of reconfiguration from S to T under TJ in G. By Lemma 4.8, for each Q_i, $i = 1, 2, \ldots, r$, $V(G) \setminus Q_i$ is a $uv-$Vertex Separator in H. And moreover, if Q_j, Q_{j+1} are two adjacent solutions in S, that is, Q_j, Q_{j+1} are independent sets in G and $|Q_j \setminus Q_{j+1}| = |Q_{j+1} \setminus Q_j| = 1$, then, by

Lemma 4.8, we have that $V(G)\setminus Q_j$ and $V(G)\setminus Q_{j+1}$ are uv-Vertex separators in H and it is clear that $|(V(G)\setminus Q_j)\setminus(V(G)\setminus Q_{j+1})| = 1$ and $|(V(G)\setminus Q_{j+1})\setminus(V(G)\setminus Q_j)| = 1$. Then the sequence

$$\langle S' = V(G) \ \setminus Q_0, V(G)\setminus Q_1, \ldots, V(G)\setminus Q_r = T'\rangle$$

is a TJ-sequence of Vertex Separator Reconfiguration from S' to T' in H. Therefore $TJ(H, S', T') =$ Yes.

Conversely suppose $TJ(H, S', T') =$ Yes. Let $\langle S' = R_0, R_1, ..., R_s = T'\rangle$ be a sequence of Vertex Separator Reconfiguration from S' to T' under TJ in H. For each $i = 1, 2, \ldots, s$ let $Q_i = V(G)\setminus R_i$. By Lemma 4.8, for each $i = 1, 2, \ldots, s$, the set Q_i is an independent set in G. Moreover, if R_j, R_{j+1} are two adjacent solutions under TJ in (H, S', T'), that is, R_j, R_{j+1} are two u, v-vertex separators in H and $|R_j \setminus R_{j+1}| = |R_{j+1}\setminus R_j| = 1$, then, by Lemma 4.8, we have that Q_j and Q_{j+1} are independent sets in G. It is clear that $|Q_j\setminus Q_{j+1}| = 1$ and $|Q_{j+1}\setminus Q_j| = 1$. Then the sequence

$$\langle S = Q_0, Q_1, \ldots, Q_s = T\rangle$$

is a TJ-sequence of Bipartite Independent Set Reconfiguration from S to T in G. Therefore $TJ(G, S, T) =$ Yes. □

Corollary 4.10. Vertex Separator Reconfiguration *is NP-hard under TAR for bipartite graphs.*

Proof. It follows from the theorem above and Theorem 4.6. □

Theorem 4.11. Vertex Separator Reconfiguration *is PSPACE-hard under TS for bipartite graphs.*

Proof. The proof of this theorem can be constructed in an absolutely analogous way to what was done in Theorem 4.9, just by observing that the change of tokens must happen just between adjacent vertices. □

4.6 Polynomial time results

In this section we present classes for which Vertex Separator Reconfiguration can be solved in polynomial time. In Subsection 4.6.1 we show that the condition of having a polynomially bounded number of minimal vertex separators is a sufficient condition for Vertex Separator Reconfiguration to be solved in polynomial time and in Subsection 4.7 we show that this condition is not necessary, giving a class of graphs that does not have polynomially bounded number of minimal vertex separators, but in which Vertex Separator Reconfiguration can be solved in polynomial time too.

4.6.1 Polynomially bounded number of minimal vertex separators class

Given a graph G with polynomially bounded number of minimal vertex separators let us construct a new graph whose vertices are minimal vertex separators in G and prove that Vertex Separator Reconfiguration can be solved in polynomial time in G under TAR/TJ.

Lemma 4.12. *Let u, v be two vertices of a graph G, $\mathbf{S}_{uv}(G)$ be the family of minimal uv-separators of G, and H_{uv} be the graph where $V(H_{uv}) = \mathbf{S}_{uv}(G)$ and $E(H_{uv}) = \{S_i, S_j \ ; \ S_i, S_j \in \mathbf{S}_{uv}(G) \text{ and } |S_i \cup S_j| \leq k\}$. For any two uv-separators S_a, S_b of G, $TAR(G, S_a, S_b, k) =$ Yes if and only if there exists a path from S'_a to S'_b in H_{uv}, where S'_a and S'_b are minimal uv-separators of G with $S'_a \subset S_a$ and $S'_b \subset S_b$.*

Proof. Suppose that $TAR(G, S_a, S_b, k) =$ Yes, let $\langle S_1, \ldots, S_r \rangle$ be a reconfiguration sequence between S_a and S_b, and let $\langle \mathcal{S}_1, \ldots \mathcal{S}_r \rangle$ be a sequence such that $\mathcal{S}_i$ is the family of all minimal uv-separators that are subsets of S_i.

Claim 4.13. *$\mathcal{S}_i \cup \mathcal{S}_{i+1}$ is a clique of H_{uv}.*

Proof. Let $A \in \mathcal{S}_i$ and $B \in \mathcal{S}_{i+1}$. Since $A \subset S_i$ and $B \subset S_{i+1}$ we have that $|A \cup B| \leq |S_i \cup S_{i+1}|$. Since $S_i \leftrightsquigarrow S_{i+1}$ under $TAR(k)$ then $|S_i \cup S_{i+1}| \leq k$. And then $|A \cup B| \leq k$. Hence A and B are adjacent in H_{uv}. □

Thus, there is a path between S'_a and S'_b in H_{uv}.

For the converse, let $\langle S'_a, S'_1, \ldots, S'_r, S'_b \rangle$ be some path between S'_a and S'_b in H_{uv}; note that since $|S'_i \cup S'_{i+1}| \leq k$, we can greedily reconfigure S'_i into S'_{i+1} without violating the cardinality constraint; by a straightforward inductive argument, we can reconfigure S'_a into S'_b and, consequently, $S_a \leftrightsquigarrow S_b$. □

Theorem 4.14. *If G is a graph with polynomially bounded number of minimal vertex separators, then* Vertex Separator Reconfiguration *problem can be solved in polynomial time under TAR/TJ.*

Proof. Let G be a graph with polynomially bounded number of minimal vertex separators. Berry et al. [4] present an algorithm which computes the set of minimal separators of a graph in $O(n^3)$ time per separator. Then the set of minimal separators of G can be generated in $O(|\mathbf{S}_{uv}(G)|n^3)$. Lemma 4.12 directly implies that it suffices to construct H_{uv} and check if there is some minimal separator contained in S_a in the same connected component of a minimal separator contained in S_b. Since H_{uv} has a number of vertices polynomial on the size of G, this algorithm runs in time polynomial on n. □

Corollary 4.15. Vertex Separator Reconfiguration *Problem can be solved in polynomial time for chordal graphs under TAR/TJ.*

4.7 Non-tame classes

Inspired by Milanič and Pivač [57], we now present a graph class that does not have a polynomially bounded number of minimal vertex separators, but in which Vertex Separators Reconfiguration Problem can be solved in polynomial time.

Milanič and Pivač [57] studied the behavior of the family of minimal vertex separators on graph classes defined by forbidden families of small induced subgraphs. Using their nomenclature, a graph class $\mathcal{G}$ is *tame* if the family of minimal vertex separators of each $G \in \mathcal{G}$, denoted by $\mathbf{S}$, has size bounded by a polynomial $p_{\mathcal{G}}$ evaluated at $|V(G)|$.

Before presenting the result of [57], let us clarify some definition and notation used. For a graph G we use $\mathbf{S}$ for the set of minimal vertex separators and $s(G)$ for the cardinality of $\mathbf{S}$.

Definition 4.16. *We say that a graph class $\mathcal{G}$ is tame if there exists a polynomial $p_{\mathcal{G}} : \mathbb{N} \to \mathbb{N}$ such that for every graph $G \in \mathcal{G}$, we have $s(G) \leq p_{\mathcal{G}}(|V(G)|)$.*

And given a family $\mathcal{F}$ of graphs, we say that a graph G is *$\mathcal{F}$-free* if no induced subgraph of G is isomorphic to a member of $\mathcal{F}$. Given two families $\mathcal{F}$ and $\mathcal{F}'$ of graphs, we write $\mathcal{F}' \trianglelefteq \mathcal{F}$ if the class of *$\mathcal{F}'$-free* graphs is contained in the class of *$\mathcal{F}$-free* graphs, or, equivalently, if every *$\mathcal{F}'$-free* graph is also *$\mathcal{F}$-free.*

For what follows, consider the graphs of Figure 4.4

Theorem 4.17. *[57] Let $\mathcal{F}$ be a family of graphs with at most 4 vertices such that $\mathcal{F} \neq \{4P_1; C_4\}$ and $\mathcal{F} \neq \{4P_1; C_4; claw\}$. Then the class of $\mathcal{F}$*-free *graphs is not tame if and only $\mathcal{F}' \trianglelefteq \mathcal{F}$ for one of the following families $\mathcal{F}'$*

- *i) $\mathcal{F}' = \{3P_1, diamond\}$,*
- *ii) $\mathcal{F}' = \{claw, K_4, C_4, diamond\}$,*

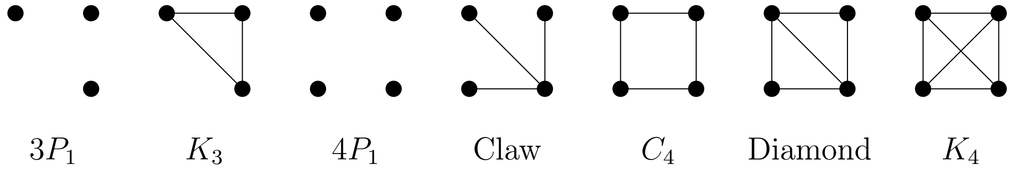


Figure 4.4: Induced subgraphs cited in this section

- *iii)* $\mathcal{F} = \{K_3, C_4\}$.

Before dealing with the reconfiguration problem for $\{3P_1, diamond\}$-*free* graphs, we present a novel characterization of the class that makes the reconfiguration question almost trivial.

Theorem 4.18. *Let G be a connected and not complete graph with at least 4 vertices, $G \neq C_5$, and let $\mathcal{F} = \{3P_1, diamond\}$. Then $G \in \mathcal{F}$-free if and only if one of the following statements holds:*

- *(i) G is the union of two cliques C_1, C_2 and has exactly one cut vertex; or*
- *(ii) $diam(G) \in \{2, 3\}$, G is the disjoint union of two cliques C_1, C_2, the edges between the cliques form a matching.*

Proof.
- ($\Rightarrow$) Let $G \in \mathcal{F}$-*free*, $G \neq C_5$. First suppose that there exists a universal vertex $v \in G$. Since G is *diamond-free* then $N(v)$ has not a P_3. Hence $N(v)$ is a disjoint union of cliques. Since G is $3P_1-$*free*, $N(v)$ is the union of at most two cliques. And since G is not complete, then G is the union of exactly two disjoint cliques. Hence G is clearly type (i).

  And now we can suppose then there does not exist a universal vertex in G and we prove by induction in number of vertices $|V(G)| = n$ of G. If $n \leq 4$ it is easy to see that G is type (ii). Suppose then the property is valid for every $G' \in \mathcal{F}$-*free*, $G' \neq C_5$, such that $|V(G')| = k$, for some $k \geq 4$. Now let $G \in \mathcal{F}$-*free*, $G \neq C_5$ be a non complete and connected graph such that $|V(G)| = k + 1$. Let v be a vertex of maximal degree in G. If $d(v) = 2$ it is trivial because in this case $G \in \{P_3, P_4, C_3, C_4\}$; then we can suppose $d(v) \geq 3$. Since G is *diamond-free* then the induced subgraph by $N(v)$ has not an induced P_3. Hence $N(v)$ is a disjoint union of cliques. Since G is $(3P_1)-$*free*, $N(v)$ is the union of at most two cliques. Let $G'' = G \setminus v$. By the induction hypothesis, $G'' = C_1 \cup C_2$ in one of cases (i) or (ii). $N(v)$ can not be one clique, otherwise either $V(G'')$ would contain a vertex with degree greater than $d(v)$ or G would be a complete graph. Then we can consider that $N(v)$ is the union of two distinct cliques, $N(v) = C_1' \cup C_2'$, both not empty. Note that C_1', C_2' are not adjacent, because G is *diamond-free*. Since C_1', C_2' are not adjacent, then we can suppose without lost of generality that $C_1' \subseteq C_1$ and $C_2' \subseteq C_2$. Since $|N(v)| \geq 3$ we can suppose again without lost of generality that $|C_1'| > 1$. We can easily see that G'' can not be of type (i) because otherwise $G = G'' \cup \{v\}$ would have an induced subgraph isomorphic to a diamond. Then G'' is type (ii). If $\exists w \in C_1 \setminus C_1'$, let $u_1, u_2 \in C_1$, then w, u_1, u_2, v is diamond. Hence $N(v) \cup \{v\} = C_1$. Since v is not universal, there exists $w \in C_2 \setminus C_2'$. If there exist $v_1, v_2 \in C_2'$ then v, v_1, v_2, w is a diamond. Hence $|C_2'| = 1$. Therefore G is type (ii).

- $(\Leftarrow)$ Follows from construction. $\square$

This family is interesting for us because it is an example of a family that does not have a polynomially bounded number of minimal vertex separators, but Vertex Separator Reconfiguration can be solved in polynomial time, as the next theorem.

Theorem 4.19. *Let G be a $(3P_1, diamond)$-free graph. Then* Vertex Separator Reconfiguration *can be solved in polynomial time in G under TAR/TJ and TS. Furthermore, under TAR/TJ it is always possible to reconfigure one separator into another.*

Proof. By Theorem 4.18, G is type (i) or type (ii). First suppose that G is type (i), that is, G is the union of two cliques Q_1, Q_2 and has exactly one cut vertex z, see Figure 4.5. Note that that the only minimal separator is the universal cut vertex z. Let S_1, S_2 be the separators which we want to reconfigure. Under TAR we can freely remove and add vertices in $S_i \setminus \{z\}$, $i = 1, 2$ and still preserve the condition of being uv-separators. Then it is easy to see that we can reconfigure S_1 into S_2 in polynomial time. Furthermore this reconfiguration is always possible. Now suppose that G is type (ii), that is, $diam(G) \in \{2, 3\}$, G is the disjoint union of two cliques Q_1, Q_2, the edges between the cliques form a matching, see Figure 4.6. Note that in this case for each minimal separator we must choose, for each edge between the cliques, exactly one of its endpoints. For the case under TAR boils down to the same analysis. For TS however, things require a bit more of work: if u, v are the vertices we want to separate and every edge between Q_1 and Q_2 has an endpoint in $\{u, v\}$, then the analysis is also equivalent to the previous case; if, on the other hand, there is at least one edge that has neither u nor v as an endpoint, we can freely move tokens between Q_1 and Q_2 through it. $\square$

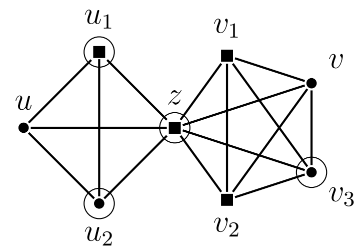


Figure 4.5: Under TAR/TJ, we can easily reconfigure the uv-separator $S_a = \{u_1, z, v_1, v_2\}$ into $S_b = \{u_1, u_2, z, v_3\}$, but not under TS since we cannot slide any token from the left clique to right without connecting u and v.

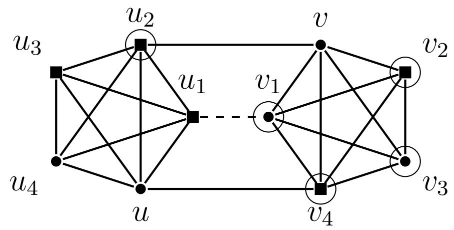


Figure 4.6: If edge $u_1v_1 \in E(G)$, under all three rules, we can easily reconfigure the uv-separator $S_a = \{u_1, u_2, u_3, v_2, v_4\}$ into $S_b = \{u_2, v_1, v_2, v_3, v_4\}$; specifically, under TS, we can use edge u_1v_1 as passageway for the tokens on the left clique. If $u_1v_1 \notin E(G)$, we cannot reconfigure S_a into S_b, since there is no way to move tokens from the left clique to the right.

Chapter 5

Conclusions

This thesis presents our studies on Vertex Separators in graphs and the results are basically arranged in the order in which they were obtained and it can be divided into two parts.

In the first part, which corresponds to Chapter 3, we studied the family of minimal vertex separators of a given class of graphs - the chordal graphs - and, based on the relationships between pairs of sets of this family, we were able to characterize these subclasses through forbidden induced subgraphs [62].

In the second part, which corresponds to Chapter 4, we studied the problem of vertex separators in graphs from the aspect of reconfiguration, that is, given two sets of vertices that separate two vertices of the graph, we study the complexity of the problem of saying whether it is possible transform one separator into another, under three specific reconfiguration rules. We showed that some problems are trivial, others have polynomial algorithms for the solution and others are still in the NP-hard class of complexity [31].

5.1 Future works

In Chapter 3 we gave a characterization by forbidden induced subgraphs for certain subclasses of chordal graphs. One way that seems natural is to seek characterizations for other classes of graphs; in particular, the first case we are studying is the class of *unichord-free* graphs. This class seems particularly interesting in this case because it has a characterization that resembles a characterization of chordal graphs: while chordal graphs can be characterized by graphs whose minimal separators are cliques (complete graphs), *unichord-free* graphs can be characterized like graphs whose minimal separators are independent sets. Although in our demonstrations we have strongly used a structure intrinsic to chordal graphs - the clique tree - which does not exist for *unichord-free* graphs, we think that we might be able to use for the *unichord-free* graphs class some of the ideas of the results we had for chordal graphs.

In the case of reconfiguration, we proved that VERTEX SEPARATOR RECONFIGURATION problem is NP-hard under TAR/TJ and it is PSPACE-hard under TJ for bipartite graphs. However, we have not studied a certificate to verify whether the problem is in NP/PSPACE or not, to prove its completeness. This is a path that can be studied from what was shown in this thesis. Besides that, as future works under TAR/TJ, a natural investigation into the complexity of the problem for different non-tame graph classes is highly desired. We note that series-parallel graph is another non tame class for which VERTEX SEPARATOR RECONFIGURATION is always possible under TAR/TJ [31]. .

Bibliography

[1] Bandelt, H. J., *Clique graphs and helly graphs*, Journal of Combinatorial Theory, Series **B 51**, (1991), 34-45.

[2] Berge, C., *Färbung von Graphen, deren sämtliche bzw. deren ungeradeKreise starr sind*, Technical report, Wissenschaftliche Zeitschriftder Martin-Luther-Universit¨at Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe **10**, (1961).

[3] Bhatt, S. N., Leighton, F.T., *A framework for solving VLSI graph layout problems* J. Comput. Syst. Sci. **28** (1984), 300-343.

[4] Berry, A., Bordat, J. P., Cogis, O., *Generating All the Minimal Separators of a Graph*, WG'99, LNCS, Springer-Verlag Berlin Heidelberg, **1665** (1999), 167-172.

[5] Bonato, A. and Nowakowski, R. J., *The Game of Cops and Robbers on Graphs*, Student Mathematical Library, **61** (2010).

[6] Bondy,J. A., and Murty, U. S. R., "Graph theory with applications", 5 ed., New Holland, New York, (1982).

[7] Blair, J. R. S. and Peyton, B., *An Introduction to Chordal Graphs and Clique Trees*, Graph Theory and Sparc Matrix Computation, IMA **56** (1992), 1-29.

[8] Bonsma, P., *The complexity of rerouting shortest paths*, Theoretical Computer Science **510** (2013), 1-12.

[9] Bonsma, P., *Independent set reconfiguration in cographs* Proc. of WG 2014, LNCS 8747 (2014), 105-116.

[10] Bonsma, P., Cereceda, L., *Finding paths between graph colorings: PSPACE completeness and superpolynomial distances*, Theoretical Computer Science **410** (2009), 5215-5226.

[11] Buneman, P., *A characterization of rigid circuit graphs*, Discrete Mathematics, **9** (1974), 205-212.

[12] Cereceda, L., van den Heuvel J. and Johnson, M. *Finding paths between 3-colorings*, Journal of graph theory, **67.1** (2011), 69-82.

[13] Cereceda, L., van den Heuvel, J., Johnson, M., *Connectedness of the graph of vertex-colourings*, Discrete Mathematics, **308** (2008), 913-919.

[14] Cerioli, M. R., Nobrega, H. and Viana, P., *Decomposition by maxclique separators*, Discrete Mathematics, **337** (2014), 119-126.

[15] Chudnovsky, M., Robertson, N., Seymour, P. and Thomas, R., *The strong perfect graph theorem*, Annals of Mathematics, **164** (2006), 51–229.

[16] De Caria, P. and Gutiérrez, M., *Introducing subclasses of basic chordal graphs*, Eletronic Notes in Discrete Mathematics, **44** (2014), 287-292.

[17] De Caria, P. and Gutiérrez, M., *On basic chordal graphs and some of its subclasses*, Discrete Applied Mathematics, **210** (2016), 261-276.

[18] Demanine, E. D. and O'Rourke, J., *Geometric folding algorithms - linkages, origami, polyhedra*, Cambridge University Press, (2007).

[19] Dirac, G. A., *On rigid circuit graphs*, Abh. Math. Sem. Univ. HamDurg, **25** (1961), 71-76.

[20] Dourado, M. C., Petiro, P., Teixeira, R. B. and Figueiredo, C. M. H., *Helly property, clique raphs, complementary graph classes, and sandwich problems*, Journal of the Brazilian Computer Society, **14** (2008), 45–52.

[21] Euler, L., *Solutio problematis ad geometriam situs pertinentis*, Comment. Acad. Sci. Imp. Petropol., **8** (1736), 128-140.

[22] Foldes, S. and Hammer, P. L., *On split graphs and some related questions*, Problèmes Combinatoires et Theorie Des Graphes, Colloques internationaux C.N.R.S., **260** (1976), 138-140.

[23] Fu, B., Oprisan, S.A. and Xu, L., *Multi-directionalwidth-bounded geometric separator and protein folding*, Int. J. Comput. Geom. Appl. **18(5)** (2008), 389-413.

[24] Fulkerson, D. R. and Gross, O. A., *Incidence matrices and interval graphs*, Pacific J. Math., **15** (1965), 5335-855.

[25] Gharibian, S. and Sikora, J., *Ground state connectivity of local hamiltonians*, Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, Proceedings, Part I, (2015), 617-628.

[26] Gavril, F., *The intersection graphs of subtrees in trees are exactly the chordal graphs*, Journal of Combinatory Theory, **B 116** (1974), 47-56.

[27] Gavril, F., *Intersection graphs of Helly families of subtrees*, Discrete Applied Mathematics, **66** (1996, 45-56.

[28] Golumbic, M. C. and Peled, U. N., *Block duplicate graphs and a hierarchy of chordal graphs*, Discrete Applied Mathematics, **124** (2002), 67-71.

[29] Golumbic, M. C., Algorithmic Graph Theory and Perfect Graphs (2nd edition), North Holland, (2004).

[30] Golumbic M. C., *Algorithmic graph theory and perfect*, Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London-Toronto, Ont., With a foreword by Claude Berge, Computer Science and Applied (1980). Mathematics.

[31] Gomes. G. C. M., Nogueira, S. H. and Santos, V. F., *Some results on Vertex Separator Reconfiguration*, Submitted, (2020).

[32] Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H., *The connectivity of Boolean satisfiability: computational and structural dichotomies*, SIAM J. Computing **38** (2009), 2330-2355.

[33] Habib, M. and Stacho, L., *Reduced clique graphs of chordal graphs*, European Journal of Combinatorics, **33** (2012), 712-735.

[34] Hajnal, A. and Surányi, J., *Uber die Ausflösung von Graphen in vollständige Teilgraphen*, Ann. Univ. Sci. Budapest, **1** (1958), 113-121.

[35] Hearn, R.A.; Demaine, E. D., *PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation*, Theoretical Computer Science, **343** (2005), 72-96.

[36] Helly, E., *Ueber mengen konvexer koerper mit gemeinschaftlichen punkter*, Jahresber. Math. Verein., **32** (1923), 175-176.

[37] Ito, T.; Demaine, E.D.; Harvey, N.J.A.; Papadimitriou, C.H.; Sideri, M.; Uehara, R.; Uno, Y. *On the complexity of reconfiguration problems*, Theoretical Computer Science, **412** (2011), 1054-1065.

[38] Ito, T.; Ono, H.; Otachi, Y., *Reconfiguration of Cliques in a Graph*, Proceedings of the 12th Annual Conference on Theory and Applications of Models of Computation, (2011) 212-223.

[39] Ito, T., Nooka, H., Zhou, X., *Reconfiguration of vertex covers in a graph*, IEIC Transactions on Information and Sistems, **E99(3)** (2016), 598-606.

[40] Johnson, W. W. and Story, W. E., *Notes on the 15 puzzle*, American Journal of Mathematics, **2** (1879), 397-404.

[41] Kamiński, M.; Medvedev, P.; Milanič, M., *Complexity of independent set reconfigurability problems*, Theoretical Computer Science, **439** (2012), 9-15.

[42] Kendall, G., Parkes, A. J., Spoerer, K., *A survey of NP-complete puzzles*, ICGA Journal, (2008), 13-34.

[43] Kennedy, W., Strictly chordal graphs and phylogenetic roots, Master Thesis, University of Alberta, (2005).

[44] Kruja, E., Marks, J., Blair, A., Waters, R., *A short note on the history of graph draqing*, 9th International Symposium, GD 2001, Vienna, Austria, Revised Papers, (2001), 272-286.

[45] Kumar, P. S. and Madhavan, C. E. V., *Minimal vertex separators of chordal graphs*, Discrete Applied Mathematics, **89 (1-3)** (1998), 155-168.

[46] Lauritzen, S. L., Graphical Models, Oxford University Press, Oxford, (1996).

[47] Leighton, F.T., *Complexity Issues in VLSI: Optimal Layout for the Shuffle-Exchange Graph and Other Network*, MIT Press, Cambridge, MA (1983).

[48] Leimer, H. G., *Optimal decomposition by clique separators*, Discrete Mathematics, **113**, (1993), 99-123.

[49] Lekkerkerker, C. G. and Boland, J. C., *Representation of a finite graph by a set of intervals on the real line*, Fundamenta Mathematicae, **51** (1962), 45-64.

[50] Lévêque, B., Maffray, F., Preissman, M., *Characterizing Path Graphs by Forbidden Induced Subgraphs*, Wiley Periodicals, Inc. J Graph Theory, **62** (2009), 369-384.

[51] Lipton, R.J. and Tarjan, R.E., *Applications of a planar separator theorem*, SIAM J. Comput. **9** (1980), 615-627.

[52] Lipton, R.J., Rose, D.J., Tarjan, R.E., *Generalized nested dissection*, SIAM J. Numer. Anal. **16(2)** (1979), 346?358.

[53] Lokshtanov, D., Mouawad, A.E, *The complexity of independent set reconfiguration on biparite graphs*, ACM Transactions on Algorithms, **15(1)** (2018).

[54] Markenzon, L., Waga, C.F.E.M., *Strictly Cordal Graphs:Characterization and Linear Time Recognition*, Eletronic Notes in Discrete Mathematics, **52** (2016), 181-188.

[55] Markenzon, L., Waga, C.F.E.M., *New results on ptolemaic graphs*, Discrete Applied Mathematics, **195** (2015), 135-140.

[56] McKee, T. A., *Independent separator graphs*, Util. Math., **73** (2007), 217-224.

[57] Milanič, M. and Pivač, N., *Minimal separators in graph classes defined by small forbidden induced subgraphs*, arXiv:1903.04534v1 [math.CO], (2019).

[58] Miller, G.L., Teng, S.H., Thurstons, W., Vavasis, S.A., *Geometric separators for finite-elementmeshes*, SIAM J. Sci. Comput. **19(2)** (1998), 364-386.

[59] Monna, c. L. and Wei, V k., *Intersection graphs of paths in a tree*, Combin. Theory Set, **B41** (1986), 141-181.

[60] Mouawad, A.E., Nishimura, N., Raman, V., *Vertex cover reconfiguration and beyond*, Algorithms and Computatioon, ISAAC 2014, (2014), 452-463 (2014).

[61] Nishimura, N., *Introduction to Reconfiguration*, Algorithms (2018), 11, 52; doi:10.3390/a11040052.

[62] Nogueira, S. H. and Santos, V. F., *Characterization by forbidden induced subgraphs of some subclasses of chordal graphs*, Electronic Notes in Discrete Mathematics, **69** (2018), 77-84.

[63] Operador Nacional do Sistema Elétrico (ONS). Site ONS. www.ons.org.br. Retreived 07 April 2020.

[64] Przytycka, T. M.; Zotenkko, E., *Graph Theoretical Approaches to Delineate Dynamics of Biological Processes*.In: MANDOIU, Ion; ZELIKOVSKY, Alexander (Eds.), Bioinformatics Algorithms: Techniques and Applications, 29-54, John Wiley and Sons, (2008).

[65] , Quilliot, A., *On the helly property working as a compactness criterion on graphs*, Journal of Combinatorial Theory, Series **A 40** (1985), 16-193.

[66] Rendly, F. and Satirov, R., *The min-cut and vertex separator problem*, Computational Optimization and Applications **69(20)** (2017).

[67] Rose, D. J., *Triangulated graphs and the elimination process*, Journal of Mathematical Analysis and Applications, **32** (1970), 597-609.

[68] Rose, D. J., Tarjan, R. E. and Lueker, G. S, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Cornput., **5** (1976), 266-283.

[69] Seker, O., Heggernes, P., Ekin, T;. Taskin, C. Z., *Linear-time generation of random chordal graphs*, arXiv:1810.13326v1 [cs.DM], (2018).

[70] Shirinivas, S.G., *Applications of graph theory in computer science an overview*, International Journal of Engineering Science and Technology, **2(9)** (2010), 4610-4621.

[71] Szwarcfiter, J. L., Grafos e algoritmos computacionais, 2 ed., Campus, Rio de Janeiro, (1986).

[72] Tarjan, R. E., *Decomposition by clique separators*, Discrete Mathematics, **55** (1985), 221-232.

[73] Tarjan, R. E.. and Yannakakis, M., *Simple linear-time algorithms t o test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM Journal on Comnputing, **13** (1984), 566-579.

[74] Vandenberghe, L. and Andersen, M. S., Chordal graphs and semide
nite optimization, Foundations and Trends in Optimization, **1(4)** (2014) 241-433.

[75] Walter, J. R., *Representations of rigid cycle graphs*, PhD thesis, Wayne State University, (1972).

[76] West, D.B.: Introduction to Graph Theory (Second edition). Prentice Hall (2001).

[77] Yamada, T. and Uehara, R., *Shortest reconfiguration of sliding tokens on a caterpillar*, Algorithms and Computation - 10th International Workshop Proceedings (WALCOM) (2016), 236-248.